KAMMER DER TECHNIK

WEITER-BILDUNG

Speicherprogrammierbare Steuerungstechnik

Lehrbrief 1

Logikverarbeitung ohne Mikrorechner

KDT-Fernkurs

KDT-Fernkurs
Speicherprogrammierbare Steuerungstechnik

Lehrbrief 1
Logikverarbeitung ohne Mikrorechner

3., unveränderte Auflage

Dr. K. Mauersberger
Dr. D. Fischer

Kammer der Technik
Präsidium

Fachverband Elektrotechnik

Sekretariatsbereich Weiterbildung

Berlin 1987

Logikverarbeitung ohne Mikrorechner : Lehrbrief / Mauersberger, Klaus ; Fischer, Dieter. - Berlin : Präsidium d. Kammer d. Technik, 1987. - 56 S. - (KDT-Fernkurs Speicherprogrammierbare Steuerungstechnik ; 1)

(C)by Eigenverlag der KDT
Clara-Zetkin-Str. 115/117, Berlin, 1086

I 12 4 Ag 238/65/87

Printed in the German Democratic Republic

Internes Lehrmaterial der Kammer der Technik
Jede Vervielfältigung - auch auszugweise - ist nur mit Genehmigung des Herausgebers gestattet.

Redaktionsschluß: 30. Dezember 1982

Inhaltsverzeichnis

		Seite
0.	Einleitung	4
0.1.	Vorbemerkungen	4
0.2.	Charakteristik speicherprogrammierbarer Steuereinrichtungen	4
1.	Grundsätzliche Arbeitsweise	9
1.1.	Serielle Abarbeitung mittels Logikoperationen	10
1.2.	Serielle Abarbeitung mittels bedingter Sprünge	13
1.3.	Taktkettenprinzip	15
1.4.	Parallele Abarbeitung mittels Tabellenverfahren	18
2.	Bauelemente und Schaltungstechnik	20
2.1.	Mittel- und hochintegrierte Bauelemente	20
2.1.1.	Speicher	20
2.1.2.	Busbausteine	23
2.2.	Operationswerk	25
2.2.1.	Operationswerk für Logikbefehle	25
2.2.2.	Operationswerk für Sprungstrukturen	28
2.3.	Steuerwerk	29
2.4.	Prozeßinterface	30
2.4.1.	Grundlagen	30
2.4.2.	Eingangsschaltungen	32
2.4.3.	Ausgangsschaltungen	34
2.5.	Zusatzbaugruppen	38
2.5.1.	Zähler	38
2.5.2.	Zeitglieder	38
3.	Programmierung	39
3.1.	Programmerstellungen nach vorgegebenem Befehlssatz	40
3.2.	Programmierung einer verbal vorgegebenen komplexen Aufgabe	46
4.	Die industrielle Steuerung PS 2000	52
Literaturverzeichnis		56

0. Einleitung

0.1. Vorbemerkungen

Die Problematik der speicherprogrammierbaren Steuerungstechnik wird in zwei Lehrbriefen behandelt.

Im vorliegenden Lehrbrief werden neben den Grundlagen die gerätetechnischen Realisierungen ohne Verwendung universeller Mikroprozessoren erklärt. Obwohl der Umfang des Lehrbriefes Beschränkungen auferlegt, wurde stets versucht, neben der Darstellung des Wirkungsprinzips, praxisorientierte Schaltungen anzugeben. Der weiterführende Lehrbrief erklärt die Unterschiede bei Einsatz von Mikroprozessoren zur Logikverarbeitung. Die grundsätzliche Funktionsweise und Programmierung dieser modernsten Elemente der Mikroelektronik ist bereits in vielen Veröffentlichungen dargestellt /1/, /2/, /3/ und wird deshalb nicht wiederholt. Auch die grundlegenden Betrachtungen zur Beschreibung, zum Entwurf und zur Realisierung verdrahteter, elektronischer Steuereinrichtungen werden als bekannt vorausgesetzt bzw. wird vielfach auf die drei Lehrbriefe "Einführung in die digitale elektronische Steuerungstechnik"/4/ verwiesen. Als zusätzliche Literatur wird /8/ und /9/ empfohlen. Dort findet man eine Fülle weiterer Literaturstellen.

0.2. Charakteristik speicherprogrammierbarer Steuereinrichtungen

Der Bereich der industriellen Steuerungen ist gegenwärtig sehr weit gefächert und reicht von der einfachen Relaissteuerung bis zum Prozeßrechner, der komplexe, räumlich verteilte analoge und digitale Vorgänge erfassen und führen kann. Eine gegenwärtig übliche Systematisierung der industriellen Steuereinrichtungen zeigt Abb. 1.

Die Klassifizierung in den ersten beiden Spalten berücksichtigt die Art und Weise, wie der Anwender das Verhalten der Steuerung dem zu steuernden Prozeß anpaßt.

In den letzten beiden Spalten wird durch die Aufteilung eine Auskunft zur Flexibilität und Bauelementebasis gegeben.

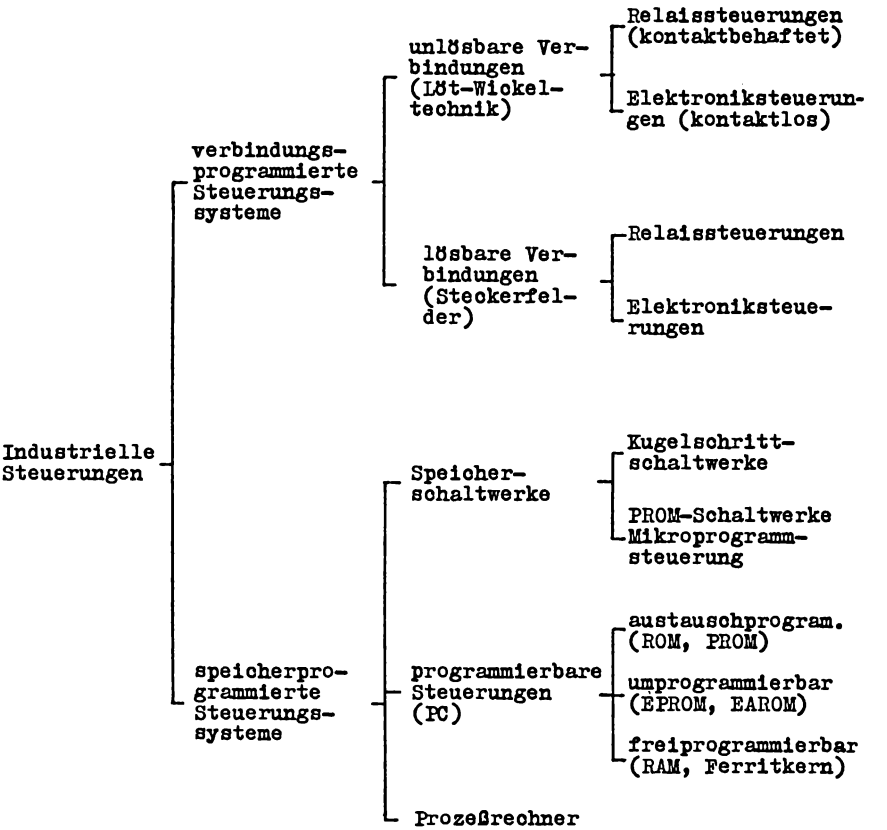


Abb. 1 Klassifikation industrieller Steuerungen

Die konventionelle Realisierung industrieller Steuereinrichtungen auf der Basis von Relais (kontaktbehaftet) und elektronischen digitalen Elementen (kontaktlos) wird heute mit dem Begriff verdrahtungsprogrammiert klassifiziert. Die eingesetzten problemneutralen Elemente (Gatter, Verstärker, Speicher) bilden mit komplexen Baugruppen oder integrierten Schaltungen (IS),

die universell nutzbare Funktionen enthalten (Adder, Multiplexer, Zähler, Register, Ein- und Ausgangsbaugruppen), ein System, das durch problemspezifische Projektierung und der daraus folgenden individuellen Verdrahtung dem jeweiligen Anwendungsfall angepaßt wird.

Jeder logischen Gleichung sind entsprechende gegenständlich vorhandene Elemente zugeordnet, die zeitlich völlig parallel arbeiten, so daß die Reaktionszeit solcher Steuerungen im wesentlichen von den Eigenzeiten der eingesetzten Elemente bestimmt wird. Bei Verwendung moderner IS ergeben sich Antwortzeiten der Steuerung auf Änderungen im Prozeß in der Größenordnung von 10^{-6} s.

Nachteilig beim Einsatz verbindungsprogrammierter Steuerungen für komplexe Aufgaben ist, daß der Aufwand und damit auch der Preis der Einrichtung mehr als proportional mit der Zahl der Ein- und Ausgänge wächst. Weiterhin muß entsprechend der problemspezifischen Verdrahtung für jede Steuerung eine eigene Dokumentation der Gerätetechnik (Hardware) erstellt werden, die außer der verbalen Beschreibung, den Logikplänen, den Stück- und Verdrahtungslisten auch Servicehinweise umfaßt. Durch diese Gegebenheiten werden die produktionsvorbereitenden Abteilungen beim Steuerungshersteller stark belastet, und diese auftragsgebundene Produktion fordert eine hohe Flexibilität in der Lagerhaltung und in den produzierenden Abteilungen.

Bei der Fertigung speicherprogrammierbarer Steuereinrichtungen PC (programmable controller) sind die genannten Nachteile nicht mehr wirksam. Der Steuerungshersteller fertigt ein problemneutrales Gerät in Serie, das auf dem Prinzip des Digitalrechners basiert. Der Befehlsvorrat und die Prozeßperipherie ist den Belangen der Steuerungstechnik angepaßt. Die aufgabenspezifische Anweisungsfolge (Software) wird vom Anwender in einem internen Speicher abgelegt.

Neben den Vorteilen beim Hersteller ergeben sich für den Anwender eine Reihe von stark verbesserten bzw. völlig neuen Gebrauchswerten, die für den progressiv wachsenden Einsatz dieser Einrichtungen mitverantwortlich sind (Abb. 2 /5/).

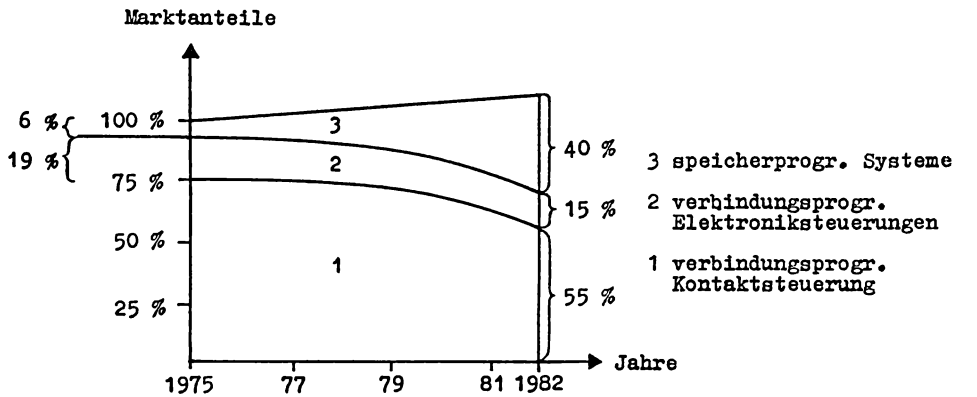


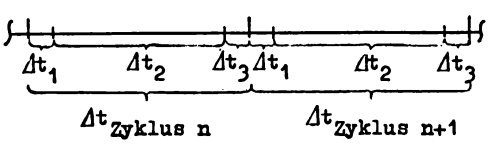
Abb. 2 Verteilung der Steuerungen an Werkzeugmaschinen

Durch die Serienproduktion und die damit einsetzbaren Fertigungstechnologien steigt die Zuverlässigkeit des Erzeugnisses. Zur einheitlichen Gerätetechnik eines Herstellers gehört es heute, daß Hilfsmittel (Programme, Schaltungstechnik) zur Überwachung der Funktionsfähigkeit und zur Lokalisierung auftretender Fehler in die Steuerung integriert sind.
Diese Maßnahmen der Eigendiagnose verbessern bei entsprechend geschultem Bedien- und Wartungspersonal die Verfügbarkeit der elektrischen Ausrüstungen beträchtlich. Besondere Aufmerksamkeit ist in unserer schnellebigen Zeit der Flexibilität zu widmen. In einigen Veröffentlichungen wird global angegeben, daß Software wesentlich einfacher, schneller und damit billiger geändert bzw. wechselnden Einsatzfällen angepaßt werden kann als Hardware. Richtig ist, daß eine komplexe Steuerungsaufgabe durch Nutzung der Programmier- und Inbetriebnahmeeinrichtungen, die es für jede PC gibt, in kürzerer Zeit und mit weniger Detailkenntnissen gelöst werden kann als der Bau einer entsprechenden verdrahteten Steuerung. Für sehr einfache Steuerungsprobleme oder minimale Änderungen einer existierenden Ausrüstung gilt die globale Aussage nicht. Beispielsweise kann sich während des Betriebes einer Anlage die Notwendigkeit ergeben, ein zusätzliches Signal in die Verriegelung eines Stellgliedes aufzunehmen.

In der verdrahteten Steuerung kann durch Auftrennen einer Leitung und Zwischenschaltung des zu berücksichtigenden Kontaktes bzw. die Beschaltung eines zusätzlichen UND-Gliedes das Problem gelöst sein. Bei speicherprogrammierbaren Steuerungen muß im Prinzip in das Anwenderprogramm nur ein zusätzlicher Befehl eingefügt werden. In den meisten Fällen wird dazu aber wieder die vollständige Programmiertechnik benötigt. Vorher muß u.U. der ganze Programmspeicher gelöscht werden, da sonst keine Änderung möglich ist. Die Vereinfachung der Manipulationen des Betreibers einer PC bei nachträglichen Anpassungsarbeiten und die Erhöhung der Transparenz der inneren Abläufe der Steuerung bei der Bearbeitung des Anwenderprogrammes sind gegenwärtig ein wichtiges Anliegen der Forschung und der Steuerungshersteller.

Gegenüber den parallelarbeitenden, verdrahteten Steuerungen vergrößert sich bei PCs sowohl die Antwortzeit auf Änderung von Prozeßzuständen (Reaktionszeit) als auch deren Streuung. Dieser Sachverhalt ergibt sich durch die schrittweise aufeinanderfolgende (sequentielle) Abarbeitung der Anweisungsfolge des notierten Anwenderprogrammes und das Festhalten (Speichern) aller Prozeßzustände während der Zeit der Programmabarbeitung (Zyklus).

Die Reaktionszeit schwankt damit zwischen 1....2 Zykluszeiten. In Abb. 3 ist zu erkennen, daß alle stochastisch auftretenden Prozeßänderungen im Zyklus n erst im Zyklus n + 1 erfaßt, bearbeitet und als neue Steuersignale ausgegeben werden können.



1 Erfassen der Prozeßzustände
2 Abarbeiten der Anweisungsfolge (Logikverarbeitung)
3 Ausgabe der Stellgrößen

Abb. 3 Zeitverhalten von PC

Die absoluten Werte der Zykluszeiten sind von den eingesetzten Bauelementen, der daraus resultierenden Taktfrequenz, der Programmlänge und -organisation abhängig. Die Zeiten liegen im

Bereich von 10^{-2} s und können deshalb für schnelle Prozesse Schwierigkeiten bringen.

Beim Preisvergleich von Steuerungsvarianten darf keinesfalls nur der Anteil der Gerätetechnik berücksichtigt werden, der in Abb. 4 prinzipiell dargestellt ist. Es sind alle Kosten der Einsatzvorbereitung, Anpassung, Dokumentation zu beachten, die sich auf die Stückzahl gleichartiger Steuerungen niederschlagen. Besonders positiv bei PCs ist, daß verschiedene Steuerstrategien und damit Algorithmen, die je nach Produkt oder Fahrweise für einen Prozeß u. U. erforderlich sind (Einrichten, Automatik, Überlastbetrieb, Anfahren, Abfahren u.ä.) bei speicherprogrammierbaren Steuerungen zwar zusätzliche, wählbare Programme, aber außer Speicherplatz keinerlei Hardwarekosten bedingen.

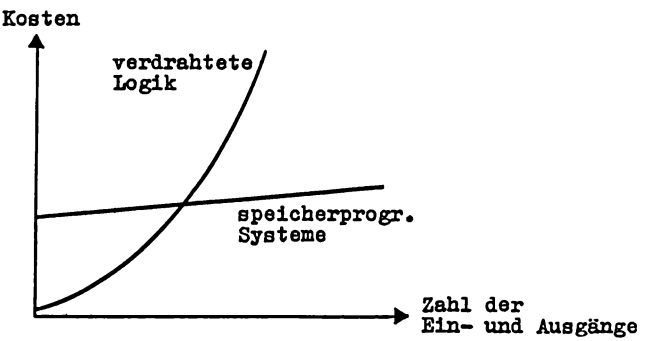


Abb. 4 Kostenverlauf von Steuerungsvarianten

1. Grundsätzliche Arbeitsweise

Die allgemeine Struktur einer PC zeigt Abb. 5.
An die Eingangsbaugruppen werden Meldesignale aus dem Prozeß und die Signale von Bedienstellen geführt. Auf den entsprechenden Leiterkarten werden die u. U. unterschiedlichen externen Pegel an die PC angepaßt und die Signale teilweise zwischengespeichert. Entsprechend den im Programmspeicher enthaltenen Befehlen werden im Logikwerk die Verknüpfungen ausgeführt (log. Gleichungen berechnet) und die Steuerwerte an die Ausgangsbaugruppen ausgegeben. Hier wird neben einer Zwischenspeicherung auch eine Anpassung an den Prozeß (Stellglieder)

mit einer Leistungsverstärkung vorgenommen.

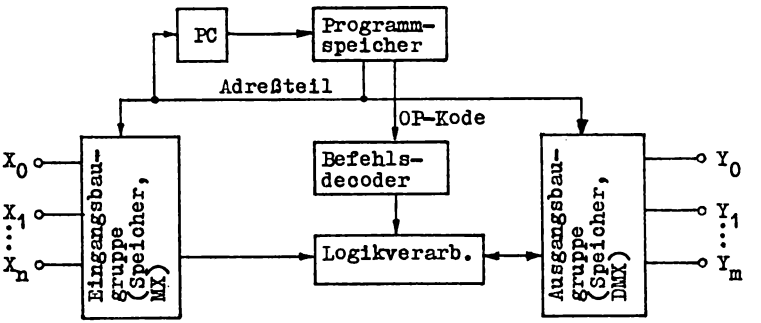


Abb. 5 Prinzipielle Struktur einer programmierbaren Steuerung

Für die Realisierung der genannten Aufgaben existieren eine Vielzahl von Strukturen und Steuerungsprinzipien, von denen die wichtigsten im folgenden näher betrachtet werden sollen.

1.1. Serielle Abarbeitung mittels Logikoperationen

Um alle logischen Steuerungsaufgaben lösen zu können, muß der Befehlsumfang einer programmierbaren Steuerung mindestens folgendes enthalten:

- Auswahl einer Eingangsvariablen und ihre Durchschaltung zum Logikwerk,
- Verknüpfung der Variablen entsprechend den Bool'schen Gleichungen,
- Ausgabe der berechneten Steuerwerte an die Ausgänge der Steuerung.

Für die logische Verknüpfung müssen mindestens die Befehle UND, ODER und NEGATION zur Verfügung stehen (vollständige Basis). Zur einfacheren Programmierung sind meist mehr Befehle vorhanden. Dabei sind besonders die Verknüpfungen negierter Variablen, Negation von Termen und Klammern für den Anwender vorteilhaft. Bei der Realisierung von Steuerungen wird der Aufruf der Variablen und deren logische Verknüpfung in einem Befehl zusammengefaßt, um Speicherplatz und Abarbeitungszeit für die

logischen Gleichungen zu sparen.

Damit besteht ein Befehl einer speicherprogrammierbaren Steuerung aus einem Operationsteil und einem Adreßteil (Abb. 6).

Im Operationsteil ist die Art der logischen Verknüpfung kodiert. Im Beispiel aus Abb. 6 sind durch 4 Bit 2^4 = 16 verschiedene Operationen verschlüsselbar.

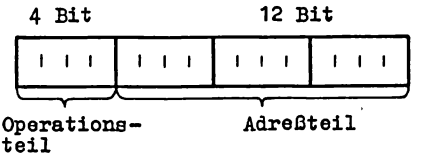


Abb. 6 Beispiel für einen Programmbefehl einer programmierbaren Steuerung

Im Adreßteil ist die Nummer (Adresse) der Eingangsvariablen eingeschrieben, mit der die Operation ausgeführt werden soll bzw. wohin der Steuerwert gebracht werden soll (Ausgabebefehle). Ist die Steuerungsaufgabe in Form von Bool'schen Gleichungen vorgegeben, kann jede Gleichung als eine Folge von Befehlen aufgeschrieben und abgespeichert werden.

Ein einfaches Beispiel soll das Verständnis erleichtern:

Eine Steuerung besitze die Befehle:

UND-Verknüpfung	(UND)
ODER-Verknüpfung	(ODER)
Ausgabe	(AUSG)
Programmende	(ENDE)

Es sollen die Gleichungen

$$X1 \wedge X2 \wedge X3 = Y1$$
$$X2 \vee X3 \vee X4 = Y2$$

berechnet werden.

Das Programm kann dann folgende Form erhalten:

Speicherzelle	Mnemonik	
00	UND	X1
01	UND	X2
02	UND	X3
03	AUSG	Y1
04	ODER	X2
05	ODER	X3
06	ODER	X4
07	AUSG	Y2
08	ENDE	(Setzen des Befehlszählers auf 00)

Mittels eines einfachen Taktzählers (Befehlszähler) werden nacheinander die Befehle aufgerufen und im Logikwerk verarbeitet. Darin ist die serielle Abarbeitung begründet. Am Ende des Programmes wird durch den Ende-Befehl der Befehlszähler rückgesetzt und das Programm dadurch zyklisch abgearbeitet.

Moderne Speicherelemente gestatten die Verarbeitung von 1000 Befehlen in 0,5 ..... 5 ms.

Wie für schnelle Prozesse die Zeit reduziert werden kann, wird später betrachtet.

In der beschriebenen Weise sind praktisch alle kombinatorischen Gleichungen programmierbar. Eine Erweiterung auf sequentielle Gleichungen ist leicht dadurch zu erreichen, daß die Verknüpfungsbefehle auf die berechneten Steuerwerte erweitert werden. Am einfachsten ist dies dadurch zu erreichen, daß diese Ausgangsgrößen auf die Eingänge zurückgeführt werden. In Abb. 7 ist dies prinzipiell dargestellt.

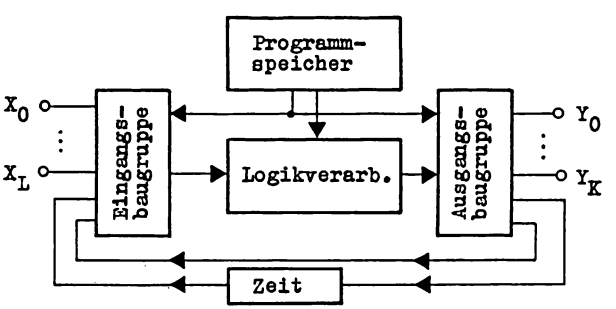


Abb. 7 Programmierbare Steuerung mit externer Rückkopplung

Ist die Abarbeitungszeit für alle Befehle wesentlich kleiner als die Reaktionszeit des zu steuernden Prozesses, so verhält sich die Steuerung trotz serieller Abarbeitung praktisch wie eine parallelarbeitende verdrahtete Steuerung. Bei sehr umfangreichen Steuerungsaufgaben wird das Programm sehr lang und damit kann die Zykluszeit so groß werden, daß die Änderung der Steuersignale nicht mehr Schritt hält mit der Änderung der Prozeßgrößen (Verletzung der Echtzeitbedingung).

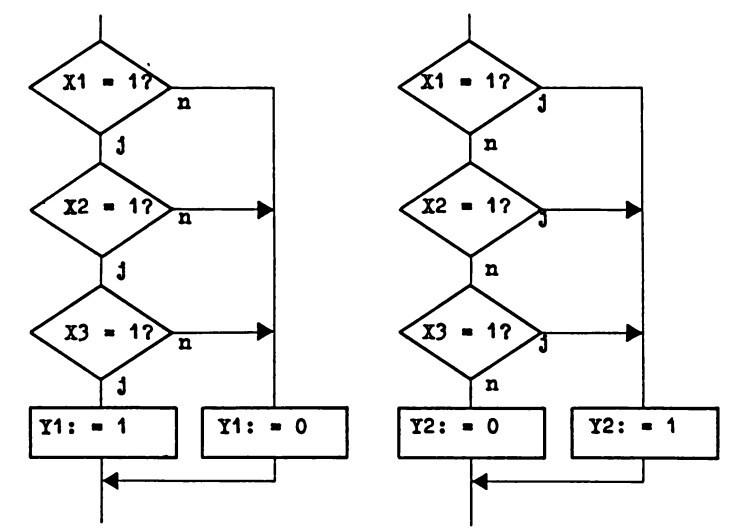
Eine Verkürzung der Zykluszeiten ist durch Einführung bedingter Sprünge in den Befehlsumfang möglich.

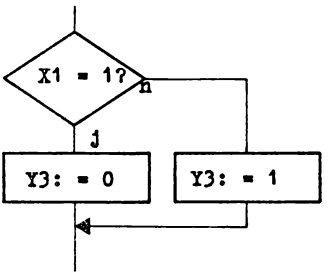
Ist in einer konjunktiven Verknüpfung eine Variable nicht erfüllt, so ist der Wert der Konjunktion dadurch schon bestimmt. Durch einen bedingten Sprung kann also gleich an das Ende der Verknüpfung gesprungen werden.

Analog ist bei einer disjunktiven Verknüpfung der Wert des Termes bestimmt, wenn eine Variable erfüllt ist. Eine Verkürzung kann auch hier durch den bedingten Sprung erreicht werden.

1.2. Serielle Abarbeitung mittels bedingter Sprünge

Der Befehlsumfang einer PC läßt sich auf bedingte Sprünge und Ausgabebefehle beschränken. Als Beweis dafür sollen die Verknüpfungen UND, ODER und NICHT als Sprungstruktur dargestellt werden (Abb. 8).





NICHT: $\overline{X1} = Y3$

Abb. 8 Programmablaufplan für UND, ODER, NICHT

Die Programmablaufpläne lassen erkennen, daß allein durch diese Anweisungen eine PC realisiert werden kann. Mit dem Befehlsspektrum aus Abb. 22 kann für die beiden Gleichungen folgendes Programm geschrieben werden:

	Speicherzelle	Mnemonik	
	00	LD	X1
	01	JMPL	M1
	02	LD	X2
	03	JMPL	M1
	04	LD	X3
	05	JMPL	M1
	06	SET	Y1
	07	JMP	M2
M1	08	RES	Y1
M2	09	LD	X1
	10	JMPH	M3
	11	LD	X2
	12	JMPH	M3
	13	LD	X3
	14	JMPH	M3
	15	RES	Y2
	16	JMP	M4
M3	17	SET	Y2
M4	18		

Es ergeben sich zwei wesentliche Vorteile:

- Der Befehlsumfang und damit die Anzahl der Bits für den Operationskode kann reduziert werden, so daß sich die Wortlänge verkürzt oder die Zahl der Adreßbit wächst. Gleichzeitig reduziert sich der Aufwand für das Logikwerk erheblich (Abb. 21).
- Im Mittelwert ergibt sich eine kürzere Zykluszeit, da durch die Sprünge nicht alle Befehle des Programmes abgearbeitet werden.

Für realisierte Steuerungen wird der Lade- und je ein Sprungbefehl zu einem Testbefehl mit Sprung vereinigt. Dieser enthält dann Eingangsadresse und Sprungzieladresse gleichzeitig.

Die bedingten Sprünge gestatten auch eine Programmverzweigung bzw. eine Art Unterprogrammtechnik bei der Programmierung. Die Programmverzweigung kann auch so genutzt werden, daß in Abhängigkeit von Prozeß- oder Bediensignalen völlig verschiedene Abarbeitungsprogramme auf einem Speicher abgelegt und wahlweise abgearbeitet werden. Bei mehreren Programmzyklen muß an die Stelle des Ende-Befehls ein unbedingter Sprung mit programmierbarem Adreßteil gesetzt werden. Ein gewisser Nachteil der Sprungbefehle ist die notwendige Adreßbreite im Befehl. Die Anzahl der Sprungadressen ist meist wesentlich größer als die Anzahl der zu adressierenden Eingangs- und Ausgangsgrößen.

1.3. Taktkettenprinzip

Durch die serielle Abarbeitung aller Steuergleichungen in jedem Zyklus können mit den bisher beschriebenen Steuerungen parallele Abläufe im Prozeß erfaßt und bedient werden. In vielen technischen Prozessen ist diese Parallelität nicht erforderlich, da nur ein Vorgang als eine Folge von Bewegungen oder anderen Schalthandlungen abläuft. Ein typisches Beispiel dafür sind einfache Manipulatoren bzw. Roboter. Der Ablauf ist oft so konzipiert, daß der Roboterarm in den einzelnen Achsen nacheinander positioniert wird und danach eine Operation abläuft. Nach Beendigung fährt der Greifer schrittweise in eine neue Position ein.

Bei derartigen Steuerungsabläufen ist eine zyklische Abarbeitung aller Gleichungen nicht notwendig.

Befindet sich die Steuerung in einem bestimmten Zustand (ein Stellglied ist in Aktion), so wird dieser erst verändert, wenn ein bestimmtes Prozeßsignal aktiv geworden ist (z.B. ein Endschalter beim Verfahren einer Achse bzw. Ablauf einer Zeitbedingung).

Man kann nun Steuerprogramme für diese Aufgabenklasse schreiben, die in jedem Zustand nur die Prozeßvariable überwachen, die durch das Taktkettenprinzip als nächstes aktiv wird.

Das gesamte Programm wird als eine Folge von kleinen Zyklen aufgebaut.

Ein Vorteil ist dabei die sehr schnelle Reaktion der Steuerung auf den Prozeß.

Das folgende Beispiel soll dies nochmals verdeutlichen.

Eine Variable X1 soll einen Antrieb im Eilvorlauf starten. Mit Erreichen des Endschalters X2 wird die Bewegung auf Schleichgang umgeschaltet. Es wird im Schleichgang verfahren, bis der Endschalter X3 aktiv wird. Mit X3 wird der Antrieb im Eilrücklauf bis X0 geführt.
Ist X1 erfüllt, so wird ein neuer Zyklus durchlaufen (Abb. 9).

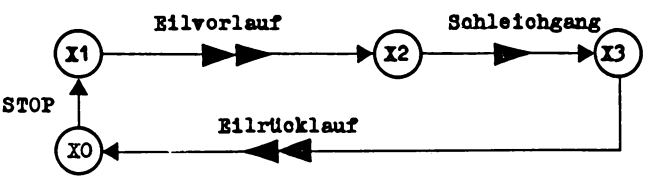


Abb. 9 Ablaufschema

Die Aufgabe kann durch den folgenden Ablaufplan formalisiert werden (Abb. 10).

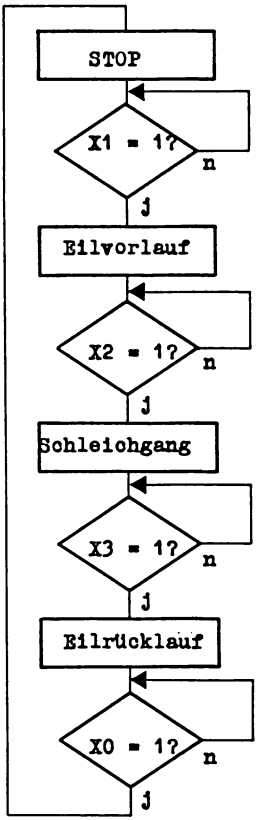
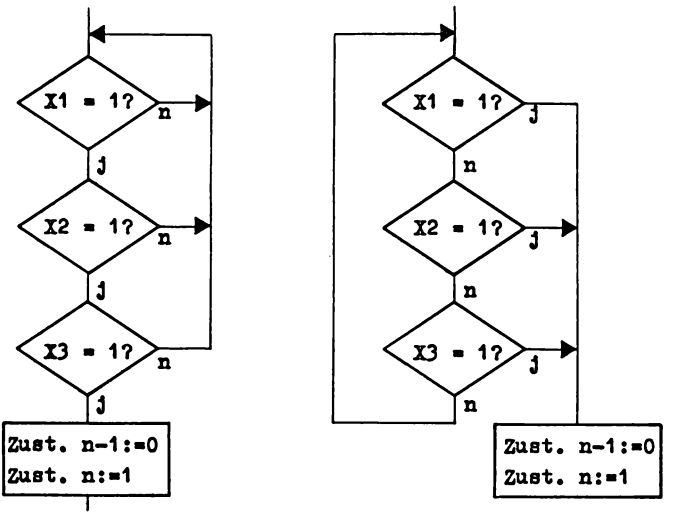


Abb. 10 Programmablaufplan zu Abb. 9

Ist die Weiterschaltbedingung nicht wie im Beispiel von jeweils nur einer Variablen, sondern einer logischen Gleichung abhängig, kann die UND- bzw. ODER-Verknüpfung nach Abb. 11 in den Ablaufplan eingefügt werden.



UND : $X1 \wedge X2 \wedge X3$ = Zustand n ODER : $X1 \vee X2 \vee X3$ = Zustand n

Abb. 11 Logische Verknüpfung in Taktkettensteuerungen

Die Verarbeitung von negierten Variablen ist möglich, läßt sich aber oft durch Hardware am Steuerungseingang umgehen. Bei diesen Taktkettensteuerungen ist der Übergang von einem Signalfolgediagramm bzw. Programmablaufgraph zur Befehlsfolge trivial. Diese zustandsbezogene Programmierung ist aber für parallelläufige, komplexe Prozesse wenig oder nicht geeignet.

1.4. Parallele Abarbeitung mittels Tabellenverfahren

Für sehr einfache Steuerungen, d.h. für eine kleine Anzahl von Eingängen, hat sich eine andere Steuerungsstruktur bewährt. In der Literatur wird sie oft als PROM-Schaltwerk oder Mikroprogrammsteuerwerk bezeichnet. Als wesentlicher Unterschied zu den bisher beschriebenen PC ist das Fehlen eines separaten Operationswerkes zu nennen. In den Speicher werden deshalb in der Programmierphase auch keine Befehle eingeschrieben. Auf den Speicherplätzen ist stets die Belegung aller Ausgangselemente parallel notiert, d.h., jeder Steuergröße ist 1 bit im Speicherwort fest zugeordnet. Für zeit- oder bedingungsabhängige Ablaufsteuerungen ähnelt die Arbeitsweise stark den elektromechanischen Kugelschrittschaltwerken. Im Unterschied zur

herkömmlichen Lösung ergeben sich aber ganz neue Möglichkeiten zur Festlegung und Variation der auszugebenden Speicherworte, indem die Folgeadresse am Speicher durch einen zusätzlichen Teil des Ausgangswortes und den Vektor der Prozeßgrößen gebildet werden kann. Für die einfachste Form der zuvor beschriebenen Taktkettensteuerung wird die Adresse über einen Befehlszähler bereitgestellt. Dieser wird in jedem Zustand durch den Signalwechsel einer ausgewählten Prozeßgröße inkrementiert. Die Auswahl besorgt ein Multiplexer, der von einem Teil des Speicherwortes gesteuert wird.

Ein anderer Speicherinhalt ergibt sich, wenn direkt die Wertetabelle eines Funktionenbündels abgespeichert wird. Die Eingangsvariablen werden in diesem Fall parallel an die Adreßeingänge des Speichers gelegt (Abb. 12).

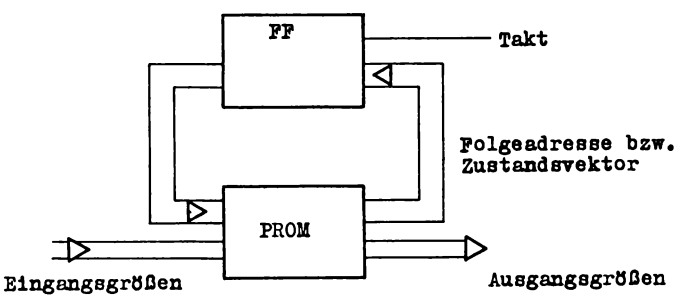


Abb. 12 Einfaches PROM-Schaltwerk

Für eine UND-Verknüpfung von 3 Variablen sind $2^3 = 8$ Kombinationen möglich. Acht Speicherplätze sind notwendig, aber nur bei der Kombination, wo alle Variablen H sind, wird im Speicherwort ein H eingeschrieben.

Man sieht, daß schon bei 10 Eingangsvariablen $2^{10} = 1024$ Speicherplätze benötigt werden.

Die Speicherbreite ist direkt proportional der Anzahl der Steuergrößen.

Sequentielle Gleichungen können nach diesem Tabellenverfahren ebenfalls bearbeitet werden. Dazu müssen durch Rückführungen die Ausgangsgrößen der sequentiellen Gleichungen an die Adressen (Eingänge) zurückgeführt werden. Für jede Speichergröße

verdoppelt sich dadurch der Speicherplatzbedarf. Diese Steuerungsvariante beschränkt sich auf einfache Anwendungsfälle.

Die Programmierung ist ebenfalls aufwendig, da die Ausgangsgrößen für jeden Speicherplatz festgelegt werden müssen.

Vorteilhaft ist dabei allerdings die sehr schnelle Abarbeitung und der übersichtliche Aufbau der Gerätetechnik wie Abb. 12 zeigt.

Weiterführende Betrachtungen und Beschreibungen von Varianten, die die aufgezeigten Nachteile teilweise vermeiden, sind in /6/ zu finden.

2. Bauelemente und Schaltungstechnik

2.1. Mittel- und hochintegrierte Bauelemente

Zu den wichtigsten Bauelementen von PCs, die ohne universelle Mikroprozessoren realisiert werden, gehören

- Halbleiterspeicher als Träger der Anweisungsfolge,
- voreinstellbare Zähler zur Adressierung der Befehle,
- Multiplexer zur Selektierung einer Prozeßgröße aus dem Eingangsvektor und deren Durchschaltung zum Logikwerk,
- Demultiplexer zum Verteilen der seriell generierten Steuergrößen an die Stellglieder des Prozesses.

2.1.1. Speicher

Obwohl gegenwärtig noch teilweise Ferritkernspeicher industriell genutzt werden, soll hier nur auf statische Halbleiterspeicher eingegangen werden. Für den Anwender ist besonders beachtenswert, daß trotz der vielen Vorteile von Halbleiterbauelementen (geringes Volumen, sehr hohe Speicherkapazität, einfache Lesbarkeit, niedriger Preis) heute für die Praxis noch kein brauchbares Bauelement zur Verfügung steht, das die Eigenschaften der Schreib- und Lesbarkeit und die Informationsspeicherung bei Energieausfall in sich vereint.

Es werden Schreib-Speicher (Informationsverlust bei Spannungsausfall) mit wahlfreiem Zugriff RAM (random access memory) und Nur-Lese-Speicher ROM (read only memory) mit wahlfreiem Zugriff ohne Informationsverlust eingesetzt, die je nach Programmier-

möglichkeit durch weitere Buchstaben klassifiziert werden. Da im ROM die Information einmalig beim Schaltkreishersteller durch eine Maske unveränderbar festgelegt wird, ist dieser Typ nur für große Serien rentabel. Der Anwender kann nur durch Austausch der IS das Programm ändern. Üblich ist für PCs der Einsatz von Lösch- und programmierbaren Festspeichern EPROM (erasable programmable ROM), so daß der Betreiber der Einrichtung mittels spezieller Programmiergeräte den Inhalt verändern kann. Für die Inbetriebnahme und Testung werden von den Steuerungsherstellern meist anschlußkompatible RAM-Speicher angeboten. Sie gewährleisten eine problemlose Informationsänderung und sind bei Verwendung der CMOS-Technologie durch Batteriestützung auch bei Abschaltung der Netzspannung vor Informationsverlust geschützt. Durch extrem niedrige Verlustleistung der CMOS-Elemente sind die Stützungszeiten > 200 h. Teilweise werden solche IS deshalb auch für den stationären Betrieb eingesetzt.

Abb. 13 zeigt das Blockbild und Anschlußschema des RAM U 202 vom VEB Funkwerk Erfurt. Es ist ein Bauelement in n-Kanal-MOS-Technologie in einem 16-poligen Gehäuse, der 1024x1-bit Worte speichern kann. Er benötigt + 5 V als Versorgungsspannung und ist an den Ein- und Ausgängen mit TTL (1 Last) kompatibel. Ein Lese- oder Schreibzyklus benötigt maximal 0,4 μs. Die Steuersignale Speicherauswahl $\overline{CS}$ (chip select) und Schreibbereitschaft $\overline{WE}$ (write enable) werden mit dem low-Pegel aktiv. Dieser Umstand wird durch den Negationsstrich über dem Signalnamen ausgedrückt. In der gesamten Mikrorechentechnik ist diese Art der Bezeichnung üblich. Unter Berücksichtigung des Dateneingangs DI (data input) ergibt sich für den Datenausgang DO (data output) die Wahrheitstabelle (Abb. 14). Ein x bedeutet, daß der logische Zustand keinen Einfluß auf den Ausgang besitzt.

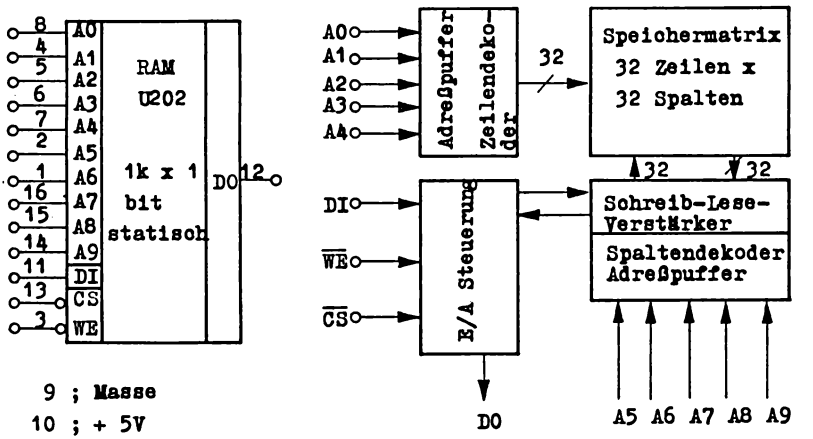


Abb. 13 Logiksymbol mit Anschlußbelegung und Blockbild U 202

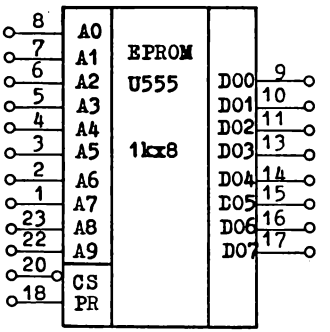
$\overline{CS}$	$\overline{WE}$	DI	DO	Funktion
H	X	X	hochohmig	inaktiv
L	L	L	L	Schreiben
L	L	H	H	
L	H	X	Inhalt der adr. Zelle	Lesen

Abb.14 Wahrheitstabelle U 202

Bei Potential H am CS (inaktiv) werden beide Transistoren der Gegentaktausgangsstufe gesperrt und damit der Ausgang hochohmig. Diese Schaltung mit 3 Zuständen (tristate) wird in allen modernen Rechner- und Steuerungskonzepten für die Realisierung von Busstrukturen benötigt. Für die Abspeicherung von Informationen mit der Wortlänge n müssen von n U 202 die Steuereingänge gleicher Bedeutung parallel geschaltet werden.

Als Beispiel für EPROM sei die IS U 555 vorgestellt. Es ist ein 1024 x 8 bit Festspeicher, der durch 100 Zyklen mit je 1 Impuls (25 V Amplitude, 1 ms Dauer am Eingang PR) je Speicherplatz elektrisch programmiert und durch intensives UV-Licht ($\approx$ 20 min) gelöscht wird. Zum Umschalten in die Betriebsart Programmieren

muß der $\overline{CS}$-Eingang auf + 12 V angehoben werden.
Der normale Lesezyklus dauert wie beim U 202 maximal 0,4 μs.
In der Abb. 15 ist das Anschlußschema wiedergegeben. Auch bei dieser IS sind die Ein- und Ausgänge direkt mit TTL-Elementen verbindbar. Es ist aber nachteilig, daß dieser Schaltkreis außer der Versorgungsspannung + 5 V noch - 5 V und + 12 V benötigt und damit erhöhte Kosten in der Stromversorgung bewirkt.



24 ; Masse
12 ; + 5V
21 ; - 5V
19 ; + 12V

Beachte: Doppelbedeutung von $\overline{CS}$

Abb. 15 Logiksymbol mit Anschlußbelegung U 555

2.1.2. Busbausteine

Bezüglich der Multiplexer und Demultiplexer sei auf /4 Abschnitt 5.1.4. und 5.1.5./ verwiesen. Eine räumlich dezentralisierte Realisierungsform von Multiplexern ergibt sich bei den bereits erwähnten Busstrukturen.

An eine Datensammelleitung sind mehrere potentielle Datensender angeschlossen, die alle durch unterschiedliche Adressen aktiviert werden. Da aber die behandelten MOS-Elemente nur den Anschluß einer TTL-Last gestatten, werden in größeren Systemen speziell entwickelte IS in Schottky low power TTL eingesetzt. Der 8216 ist ein 4 bit bidirektionaler Treiber. Abb. 16 zeigt das Anschlußschema und Blockbild. Mit dem Steuersignal Dateneingabeermächtigung $\overline{DIEN}$ (data input enable) wird die Signalflußrichtung festgelegt.

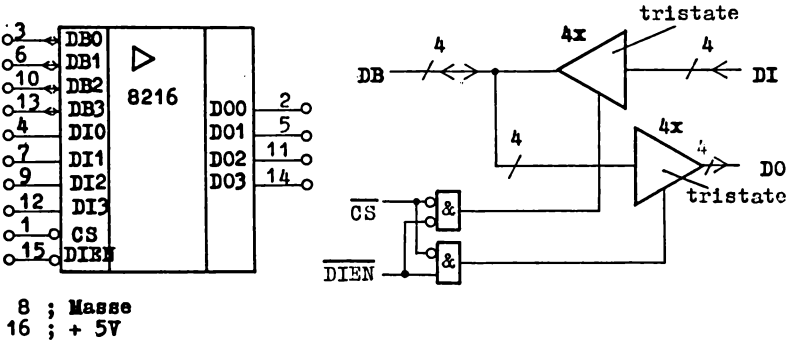
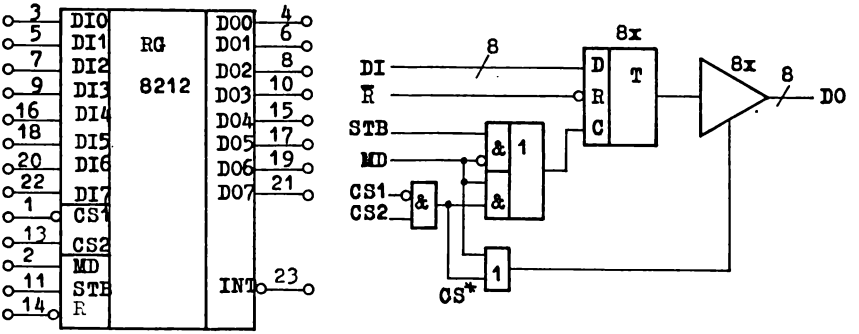


Abb. 16 Logiksymbol mit Anschlußbelegung und Blockbild 8216

Beachte: Die Begriffe Eingabe und Ausgabe sind in Einrichtungen mit zentraler Verarbeitungseinheit stets auf diese als Bezugspunkt benutzt.

Der Schaltkreis 8212 Abb. 17 ist ein unidirektionales, 8 bit paralleles, taktzustandsgesteuertes Register mit tristate Verhalten. Durch einen zusätzlichen Eingang zur Wahl der Betriebsart MD (mode) kann der Schaltkreis an verschiedene Einsatzfälle angepaßt werden. Bei MD = H ist der Ausgang immer aktiv,und während CS* = H ist der Logikpegel an den Ein- und Ausgängen identisch. Das Auslösesignal STB (strobe) besitzt keine Wirkung. Bei MD = L wird in den Speicher die Information nur bei STB = H eingeschrieben. Der aktive Zustand der Ausgänge wird nur durch CS* = H bewirkt. Das Unterbrechungssignal $\overline{INT}$ (interrupt) wird hier nicht erklärt, da es nur in Verbindung mit universellen Prozessoren benötigt wird.



12 ; Masse
24 ; + 5V

Abb. 17 Logiksymbol mit Anschlußbelegung und Blockbild 8212 (ohne INT-Steuerung)

2.2. Operationswerk

Die Zentraleinheit von Rechnern und speicherprogrammierbaren Steuerungen läßt sich wie jeder Automat theoretisch in ein Steuerwerk und ein Operationswerk aufteilen. Bei universellen Mikroprozessoren (U 808, U 880 u.ä.) sind beide Teile praktisch in einem hochintegrierten Bauelement vereinigt. Die eigentliche Logikeinheit von PCs wird durch die kleine Menge ausführbarer Befehle schaltungstechnisch so einfach, daß nur wenige Hersteller (z.B. Motorola) dafür einen speziellen Schaltkreis entwickelt haben.

2.2.1. Operationswerk für Logikbefehle

In Abb. 18 ist der vollständige Logikplan eines Operationswerkes dargestellt, das die Befehlsliste (Abb. 19) realisiert. Bemerkenswert ist das Vorhandensein eines getrennten UND-Akkus und eines ODER-Akkus. Dadurch können sehr einfach Gleichungen in disjunktiver oder konjunktiver Form programmiert werden. Da auch Befehle für die negierten Werte von beiden Akkumulatoren vorhanden sind, lassen sich direkt NAND- bzw. NOR-Terme bilden oder läßt sich das negierte Gleichungsergebnis abspeichern.

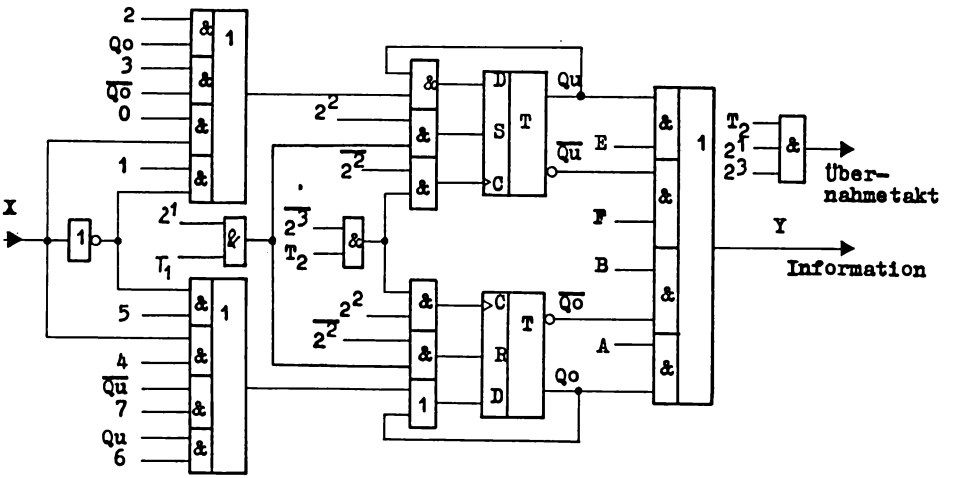


Abb. 18 Operationswerk für Logikbefehle

Mnemonik	OP	Kodierung 2^3 2^2 2^1 2^0	Hexa	Erklärung
UND X		0 0 0 0	0	UND eines adressierten Einganges
UND $\overline{X}$		0 0 0 1	1	UND eines adr. neg. Einganges
UND Qo		0 0 1 0	2	UND des ODER-Ergebnisses
UND $\overline{Qo}$		0 0 1 1	3	UND des neg. ODER-Ergebnisses
ODER X		0 1 0 0	4	ODER eines adressierten Einganges
ODER $\overline{X}$		0 1 0 1	5	ODER eines adr. neg. Einganges
ODER Qu		0 1 1 0	6	ODER des UND-Ergebnisses
ODER $\overline{Qu}$		0 1 1 1	7	ODER des neg. UND-Ergebnisses
SP Qo		1 0 1 0	A	Speichern des ODER-Ergebnisses
SP $\overline{Qo}$		1 0 1 1	B	Speichern d. neg. ODER-Ergebnisses
SP Qu		1 1 1 0	E	Speichern des UND-Ergebnisses
SP $\overline{Qu}$		1 1 1 1	F	Speichern d. neg. UND-Ergebnisses

Abb. 19 Befehlsliste des Operationswerkes aus Abb. 18

Für die Erklärung der Funktionsweise wird vom Anfangszustand der Akkus (Qu = H, Qo = L) ausgegangen, der sowohl beim Einschalten als auch nach den Befehlen 2, 3, 6, 7, A, B, E oder F erreicht wird. Am Eingang X steht die durch die Adresse vom Multiplexer ausgewählte Prozeßvariable oder ein zuvor errechnetes Zwischenergebnis aus einem adressierbaren Datenspeicher (z.B. 1x U 202) an.

Über die Rückführung von Qu bleibt das Tor am D-Eingang so lange geöffnet, bis zum ersten Mal bei einem UND-Befehl ein L-Signal verarbeitet wurde. Danach bleibt das Ergebnis stets L. Für den Qo gilt die Beschreibung prinzipiell. Zum Unterschied ist der Anfangswert Qo = L. Die Befehlsaufrufe 0....7 bzw. A, B, E, F werden von einem Dekoder geliefert, während die Signale 2^1, 2^2, 2^3 direkt die Bits des Operationskodes darstellen. Das Taktsignal T 2 wird vom Steuerwerk stets nach Erhöhung des Befehlszählers mit T 1 und dem Ablauf aller Signalzeiten geliefert (Abb. 20).

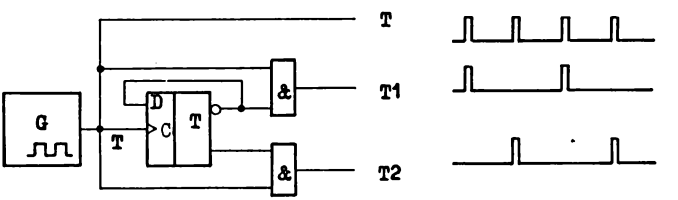


Abb. 20 Schaltung für 2 nichtüberlappende Takte

Zum Verständnis der Funktionsweise der Ausgangsschaltung ist zu beachten, daß Y parallel an allen D-Eingängen aller Ausgangsspeicher anliegt und nur das Übernahmesignal durch die Adresse an einen einzigen Ausgangsspeicher vom Demultiplexer als Takt gegeben wird (Abb. 25).

Beachte: Um die Funktionsweise des Operationswerkes deutlicher zu machen, wurde keine Einschränkung durch die Bauelementebasis berücksichtigt.

2.2.2. Operationswerk für Sprungstrukturen

Rein seriell arbeitende Ablaufsteuerungen sind einfacher beschreibbar und einfacher schaltungstechnisch zu realisieren. In der Abb. 21 ist ein Logikplan gezeigt. Die ausführbaren Befehle sind in Abb. 22 zusammengefaßt.

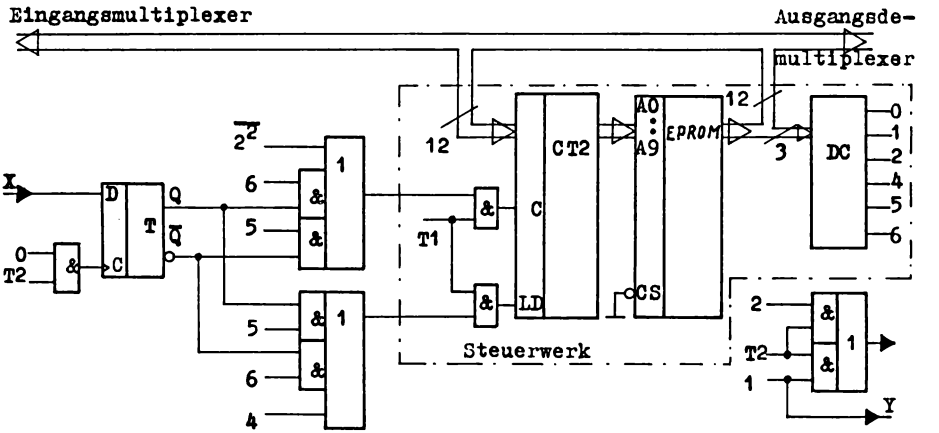


Abb. 21 Operationswerk und Steuerwerk für Sprungstrukturen

Mnemonik		OP Kodierung 2^2 2^1 2^0			Hexa	Erklärung
LD	X	0	0	0	0	Laden eines adressierten Eingangs
SET	Y	0	0	1	1	Setzen eines adressierten Ausgangs
RES	Y	0	1	0	2	Rücksetzen eines adressierten Ausgangs
JMP	ADR	1	0	0	4	unbedingter Sprung
JMPH	ADR	1	0	1	5	Sprung bei high
JMPL	ADR	1	1	0	6	Sprung bei low

Abb. 22 Befehlsliste des Operationswerkes aus Abb. 21

Die in der Spalte "Hexa" eingetragenen Ziffern stimmen wiederum mit den Ziffern der Steuersignale im Logikplan überein. Man erkennt, daß der momentane Zustand (H;L) einer aufgerufenen Variablen stets das Flip-Flop zum Zeitpunkt T 2 überschreibt und das Operationswerk den Ausgang Y nicht direkt beeinflußt. Über die Befehle 5 oder 6 kann aus der Belegung Q der weitere Programmablauf abhängig gemacht werden. Bei den genannten Anweisungen wirkt das Operationswerk auf das Steuerwerk zurück. Dessen Befehlszähler wird zum Zeitpunkt T 1 entweder inkrementiert oder mit dem Adreßteil des Befehles überschrieben (LD = H). Für die Befehle Ausgabe (1,2) gelten die Erklärungen aus Abschnitt 2.2.1.

2.3. Steuerwerk

In Abb. 21 waren schon die wesentlichen Komponenten der Steuerwerke von PCs zu sehen. Zu den Aufgaben des Steuerwerks gehört die Erzeugung von Grundzuständen, die zeitlich richtige Öffnung und Schließung von Datenwegen über Ansteuerung entsprechender Tore und die Auswertung von Zuständen des Operationswerkes oder des Operationskodes für die Beeinflussung des Ablaufes. Auch beim Operationswerk nach 2.2.1. gibt es mindestens einen Sprungbefehl, um die Anweisungsfolge des Anwenders als Zyklus zu schließen. Beide vorgestellten Operationswerke arbeiten in zwei Phasen.

T 1: Bildung einer neuen Adresse, Abwarten der Zugriffszeit des Speichers, Dekodierung des neuen Operationskodes, eventuell Durchschalten des Multiplexers oder Demultiplexers an Ein- bzw. Ausgang.

T 2: Einspeicherung der gefundenen Variablenbelegung, Bereitstellung des Ergebnisses im Operationswerk für die neue T 1 Phase.

Wenn nur taktflankengesteuerte Trigger eingesetzt werden, kann die Vorder- und Rückflanke eines einfachen Taktes benutzt werden. In der Praxis ist dieser Fall meist nicht erfüllbar (taktzustandsgesteuerter Ladeeingang beim Befehlszähler). Zwei Auswege sind möglich. Durch eine geeignete Schaltung /4 Abschnitt 5.3.2./ wird beim Zustandswechsel des Taktes ein kurzer Impuls erzeugt.

Diese Methode hat den Nachteil, daß die Funktion nie statisch nachweisbar ist. Eine Möglichkeit mit etwas größerem Schaltungsaufwand zeigt Abb. 20.

Der Taktgenerator arbeitet mit der doppelten Frequenz von T 1 und T 2. Durch unsymmetrischen Takt läßt sich die H-Phase von T 1 und T 2 beeinflussen.

Für Servicezwecke kann vom Taktgenerator auf Handtakt umgeschaltet werden und in jeder Phase statisch gemessen werden.

2.4. Prozeßinterface

2.4.1. Grundlagen

Den Schnittstellen zwischen digitaler elektronischer Informationsverarbeitung und den Bereichen der Prozeßmeß- und Prozeßstelltechnik wird in der Gegenwart immer mehr Aufmerksamkeit geschenkt. Durch die konstruktive Gestaltung, die eingesetzten Bauelemente und die Dimensionierung wird ganz wesentlich die Zuverlässigkeit, der Preis und die Servicefreundlichkeit der Steuerungseinrichtung beeinflußt. Die grundsätzlichen Forderungen an das Prozeßinterface sind für verdrahtete und speicherprogrammierbare industrielle Steuerungen gleich. Bei den unterschiedlichsten Einsatzfällen und Umgebungsbedingungen muß das Interface weitgehend zerstörungssicher gestaltet werden und sollte auch bei stark gestörten Signalen die Information fehlerfrei übertragen. Treten trotz aller Maßnahmen Bauelementeausfälle auf, so muß der Prozeß unter allen Umständen in den gefahrlosen Zustand gebracht werden. Nach Möglichkeit sind die Fehler durch die Einrichtung selbst zu diagnostizieren, dem Bedienpersonal anzuzeigen und entsprechende Gegenmaßnahmen einzuleiten. Aus Gründen der Störsicherheit bei der Signalübertragung in industriellen Prozessen und zur Gewährleistung der Funktionsfähigkeit von kontaktbehafteten peripheren Elementen hat sich national und international ein Pegel von + 24 V Gleichspannung durchgesetzt. Da in den Einheiten der Informationsübertragung aus Gründen der TTL-Kompatibilität ein Pegel von + 5 V üblich ist, muß die Interfacebaugruppe ein- und ausgangsseitig die erforderliche Pegelanpassung realisieren. Neben dem Spannungspegel ist insbesondere bei Ausgangssignalen zu den Stellelementen auch das Leistungsniveau anzupassen. Beide Probleme gehören

zur statischen Anpassung des elektrischen Signales.

In einigen Fällen ist auch eine dynamische (zeitliche) Anpassung der internen und externen Einrichtungen erforderlich. Bei der seriellen Abarbeitung speicherprogrammierbarer Steuerungen und der dadurch bedingten Zykluszeit dieser Einrichtungen müssen impulsförmige Eingangssignale verlängert oder bis zur Abfrage gespeichert werden und u. U. manuell erzeugte Signale in ihrer Länge begrenzt werden /4 Abschnitt 5.3.3./.

Zum Stand der Technik gehört gegenwärtig die visuelle Anzeige des Signalpegels jedes Ein- und Ausganges mittels Lumineszenzdioden.

Viele Hersteller bieten heute auch Interfacebausteine an, die durch Optokoppler eine vollständige galvanische Trennung gewährleisten, so daß örtliche Verschiebungen des Bezugspotentials keine Signalverfälschungen hervorrufen können.

Neben der erwähnten dynamischen Anpassung gibt es bei speicherprogrammierbaren Steuerungen weitere Besonderheiten bei den Interface-Kartenbaugruppen, die sich aber fast alle aus der seriellen Arbeitsweise des zentralen Verarbeitungsteiles ergeben. Da jeder Steuerungsausgang nur für den Bruchteil der Zykluszeit bedient wird, muß jedes generierte Ausgangssignal für die Zeit zwischen zwei Bedienungen gespeichert werden. Meist werden D-FF für diese Zwecke eingesetzt. Auch bei den Eingangsschaltungen ist die Speicherung aller Prozeßgrößen zu einem definierten Zeitpunkt (Befehl) kaum zu umgehen, da ein beliebiger Eingang in mehreren Bool'schen Funktionen auftreten kann und ohne Speicherung ein Signalwechsel zwischen den mehrmaligen Abfragen in einem Zyklus zu Widersprüchen in der Logik führen kann. Am auffälligsten sind die Abweichungen gegenüber verdrahteten Steuerungen, die sich aus dem Einsatz der konstruktiv vorteilhaften Bus-Konzepte ergeben. Jede Kartenbaugruppe muß adressierbar gemacht werden und alle nichtadressierten Sender müssen inaktiv bleiben. Man erreicht durch diese Busgestaltung eine sehr gute Modularität (Auf- und Abrüstbarkeit der Ein- und Ausgänge) und eine sehr einfache anwenderunabhängige Rückverdrahtung, die in modernen Anlagen als gedruckte Leiterplatte (billig) ausgeführt wird.

2.4.2. Eingangsschaltungen

Für die elektronische Ausrüstung an Be- und Verarbeitungsmaschinen sind die Sicherheitsforderungen in der TGL 200-0655 Blatt 3 festgehalten, wobei diese TGL inhaltlich mit der IEC 204-3 übereinstimmt. Für Eingangssignale der Steuerung ist festgelegt, daß weder durch Leitungsbruch noch durch Kurzschluß einer Signalleitung mit Bezugspotential ein ungewolltes Einschalten eines gefährlichen Zustandes erfolgen darf und durch die genannten Fehler in der Peripherie bzw. Verbindungstechnik auch kein gewolltes Ausschalten eines gefährlichen Zustandes verhindert werden darf. Werden die Signale in der Peripherie durch Kontakte erzeugt, so müssen für "Aus" Öffner und für "Ein" Schließer zur Anwendung kommen. In der Literatur /4 Abschnitt 5.3.2./ sind dazu bereits Schaltungen auf TTL-Basis angegeben. Von unserer Bauelementeindustrie sind inzwischen für diese Zwecke spezielle Schaltkreise entwickelt worden. Der Schaltkreis D 410 enthält 3 UND-Gatter, denen jeweils ein Verzögerungsglied und ein Schmitt-Trigger nachgeschaltet ist /6/. Im Gegensatz zu TTL-Elementen sind die Eingänge hochohmig und high-aktiv und die Schaltschwelle mit + 7,5 V festgelegt. Da die Eingänge direkt für 24 V Pegel vorgesehen und dauernd für $\pm$ 35 V, kurzzeitig aber bis $\pm$ 220 V über den vorgeschriebenen 5,6kΩ Widerstand spannungsfest sind, ist genügend Sicherheitsreserve vorhanden. Der vorhandene z-Eingang der IS ermöglicht auf einfachste Weise eine Pegelanpassung am Ausgang zu CMOS bzw. low power TTL, indem das Potential am z-Eingang die maximale Ausgangsspannung bestimmt.

Der Schaltkreis D 412 hat die gleiche Eingangsstufe, aber der Ausgang besitzt tristates Verhalten und ist damit direkt busfähig.

In Abb. 23 ist eine Schaltung zur Eingabe von 8 Prozeßsignalen mittels 3 D 410 und einem Busregister 8212 dargestellt. Um alle Karten identisch aufbauen zu können, läßt sich die Adresse der Karte (des Kanales) vom Anwender durch Wickeln von Lötbrücken oder Einstellen von Schaltern am Eingang des Dekoders frei wählen. Erkennt man die Adresse in einem Befehl, werden alle 8 Eingangssignale durch Aktivierung des 8212 auf den Bus gelegt, so daß im zentralen Verarbeitungsteil noch ein 8 auf 1 Multiplexer durch 3 weitere Adreßbit eine binäre Variable

selektiert. Die parallele Bereitstellung von 8 bit durch die Eingangskarte spart den Multiplexer je Eingangskarte ein und ist besonders bei Einsatz von universellen Mikroprozessoren als zentralem Verarbeitungsteil vorteilhaft (Lehrbrief 2).

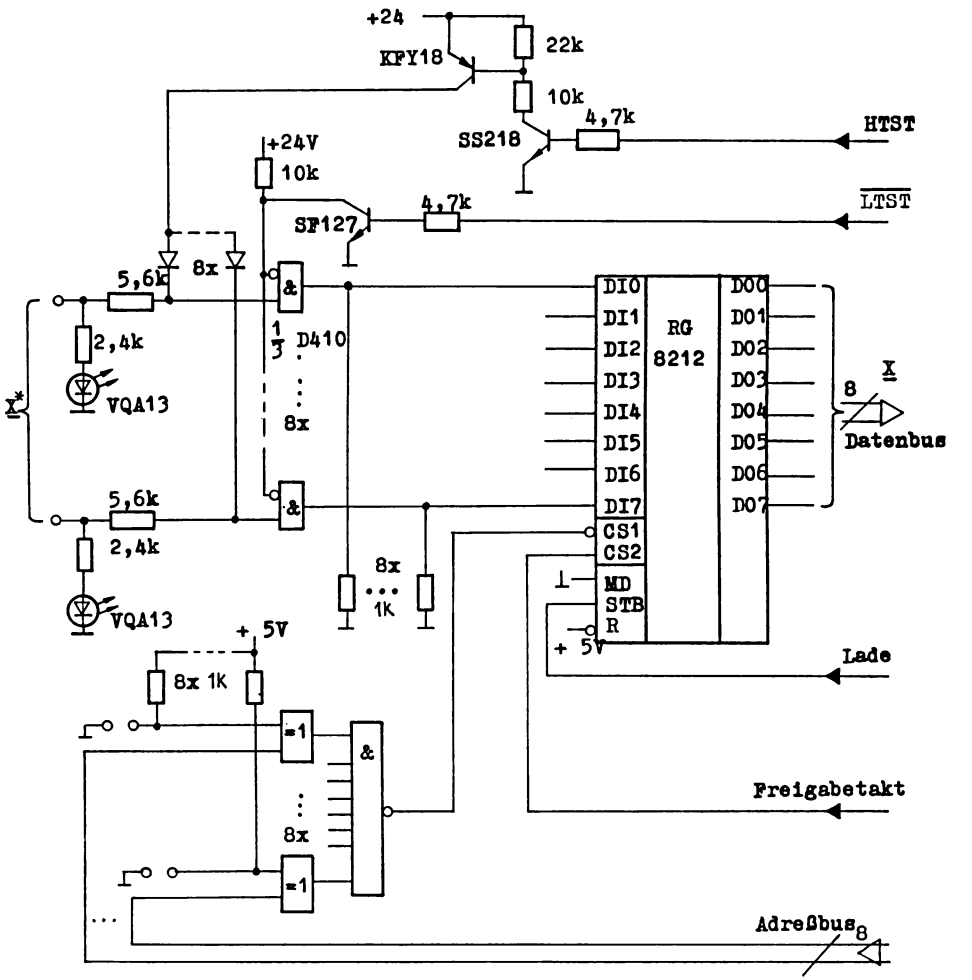


Abb. 23 Eingabeschaltung für 8 bit

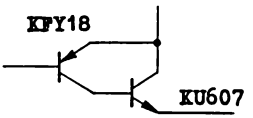


Abb. 25 Erweiterung für T 3 bei großen Strömen

Im Abschnitt 2.4.1. wurde begründet, daß für alle PC, außer bei parallelen PROM-Schaltwerken, die Ausgangssignale gespeichert werden müssen. Die eingesetzten Speicherelemente und die Ansteuerschaltungen sind eng mit der Arbeitsweise des zentralen Verarbeitungsteiles verknüpft. Um die Busstruktur beizubehalten, die für die Eingänge beschrieben wurde, werden die im Bearbeitungszyklus nacheinander entstehenden Ausgangssignale in einen bitorganisierten RAM eingeschrieben. Am Schluß der Generierung aller Ausgangssignale werden die Ergebnisse nach Serien-Parallel-Wandlung wortweise (meist 8 bit) über den Bus in die Register der Ausgangskarten übernommen. Diese Organisation bringt den zusätzlichen Vorteil, daß alle Ausgangssignale während der Bearbeitung der Gleichungen durch den zentralen Verarbeitungsteil direkt vom RAM lesbar sind. Im Bild 26 ist eine Schaltung für 8 Bit Wortbreite angegeben.

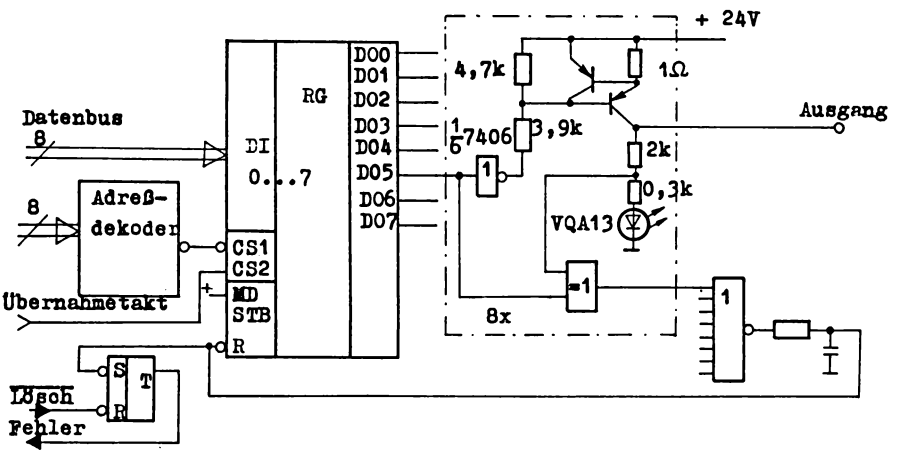


Abb. 26 Ausgangsschaltung (wortorganisiert)

Der Kurzschlußschutz wird im Gegensatz zu Abb. 24 auf eine ganz andere Weise gewährleistet. Die eingesetzten Antivalenzgatter vergleichen ständig die am Registerausgang bereitstehenden Signale mit den Signalen am Ausgangstransistor. Ist keine Übereinstimmung vorhanden, so wird der Registerinhalt gelöscht.

Das Fehlersignal wird für die Laufzeit im Ausgangsverstärker durch das RC-Glied unwirksam. Die Überwachungsschaltung erkennt außer einem Kurzschluß der Ausgangsleitung gegen Bezugspotential auch einen Kurzschluß des Ausgangstransistors, der das Abschalten verhindern würde. Das gespeicherte Fehlersignal kann in der Steuerung ein Fehlerprogramm starten. Im einfachsten Fall wird über das Signal die externe Stromversorgung + 24 V für die fehlerhafte Karte abgeschaltet.

Bei Steuerungen mit einer Bitausgabe ist dem Demultiplexer besondere Aufmerksamkeit zu schenken. In den Abb. 18 und 21 werden jeweils 2 Signale bereitgestellt. Neben dem Ausgang Y wird ein Übernahmesignal für das Ausgangsregister erzeugt. Die Abb. 27 zeigt die Ansteuerschaltung für eine Karte mit Bitausgabe. Ein Nachteil ist der niedrige Integrationsgrad (2 D - FF je IS) für Speicher mit getrenntem Takteingang.

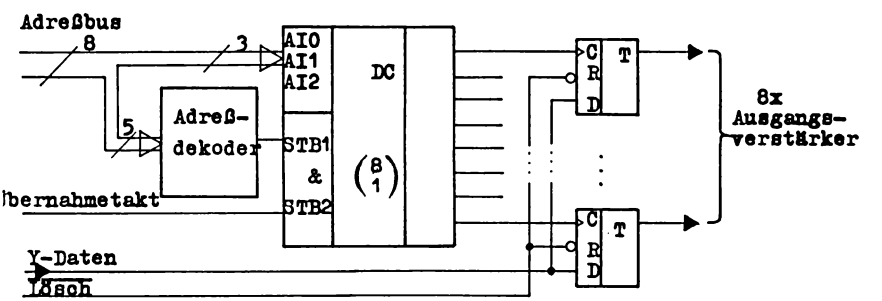


Abb. 27 Ausgangsschaltung (bitorganisiert)

2.5. Zusatzbaugruppen

2.5.1. Zähler

Eine oftmalige Aufgabe für industrielle Steuerungen ist das Zählen von Ereignissen. Zählfunktionen werden für die bedingungs- oder zeitabhängige Auslösung von Vorgängen benötigt. Mit dem im Abschnitt 2.2.2. vorgestellten Befehlsspektrum und lesbaren Zwischenspeichern (RAM) können Zähler als Befehlsfolge realisiert werden. Die Programmierung ist durch die Bitverarbeitung und das Fehlen von decrementierbaren Registern allerdings recht aufwendig. Außerdem muß die Zeit zwischen 2 Zählimpulsen unter allen Umständen größer als die doppelte Zykluszeit der PC sein, so daß in der Praxis nur Frequenzen bis $\approx$ 100 Hz gezählt werden können. Auf Softwarelösungen wird deshalb erst im Lehrbrief 2 eingegangen.

Vielseitig einsetzbar sind Leiterkarten mit voreinstellbaren Rückwärtszählern, die bei der Kopplung mit der PC eine einfache Systemlösung ermöglichen.

Die PC wertet in ihrem Programm den Nulldurchgang bzw. Übertrag des Zählers aus und liefert an einem Ausgang ein Signal, wenn der Zähler wieder neu auf seinen Anfangswert gesetzt werden soll. Der Zahlenwert zum Laden des Zählers ist oftmals über Ziffervorwahlschalter vom Anwender leicht zu beeinflussen. Unter Umständen kann der Impulseingang des Zählers über einen weiteren Ausgang der PC getort werden. Zur Schaltungstechnik der Zähler wird auf /4 Abschnitt 5.2./ verwiesen.

2.5.2. Zeitglieder

Besonders wichtig für industrielle Steuerungen ist die Realisierung von Ein- und Ausschaltverzögerungen. Bei einfachen PC wird diese Aufgabe extern gelöst. Das Prinzip einer Einschaltverzögerung zeigt Abb. 28.

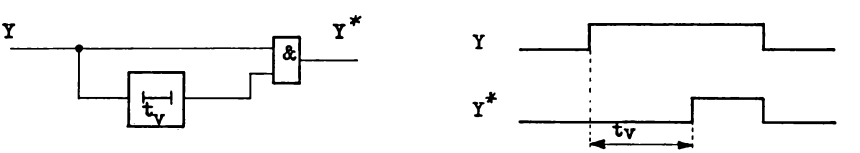


Abb. 28 Prinzip einer Einschaltverzögerung

Im einfachsten Fall wird an einen Ausgang der PC ein analog oder digital arbeitendes Zeitglied angeschaltet und dessen Ausgang mit einem Steuerungseingang verbunden. Die UND-Verknüpfung aus Abb. 28 wird dann im Programm realisiert. Jede Rückführung benötigt aber einen Eingang. Auf dem Niveau von + 24 V läßt sich die Verzögerung mit der IS D 410 besonders einfach gestalten /7/.

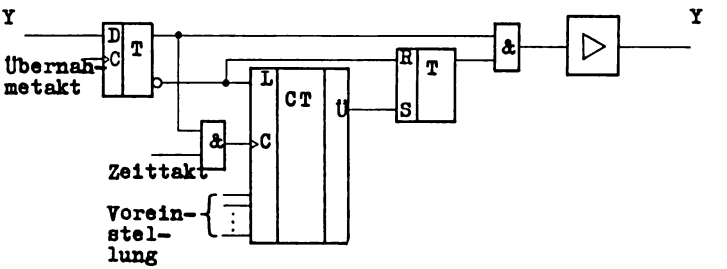


Abb. 29 Einschaltverzögerung mit digitaler Zeitvorgabe

In Abb. 29 ist ein Lösungsvorschlag für eine digitale Variante angegeben, die zwischen Ausgangsspeicher und Leistungsverstärker von Abb. 27 wie ein Element mit analoger Zeitbeeinflussung eingefügt werden kann. Die digitale Lösung ergibt einen höheren Aufwand bei wesentlich besserer Reproduzierbarkeit der Werte.

3. Programmierung

Nachdem in den vorangegangenen Kapiteln die Realisierungsmöglichkeiten dargelegt wurden, wollen wir uns nun der Programmierung von speicherprogrammierbaren Steuerungen widmen. Die Steuerungsaufgabe ist dabei in eine Befehlsfolge umzusetzen. Dies soll möglichst schnell und einfach durchführbar sein. Damit wird erreicht, daß die bereits erwähnte Flexibilität der speicherprogrammierbaren Steuerungen wirksam werden kann. Je weniger Spezialkenntnisse zur Programmierung erforderlich sind, um so schneller werden sich diese Steuerungen in der Praxis einführen. Programmierbare Steuerungen haben im Normalfall nur logische Operationen auszuführen. Die wenigen dafür notwendigen Grundoperationen drücken sich auch in dem geringen Befehlsumfang

der Steuerungen aus (4....20 Befehle).

Den Befehlen sind die steuerungstechnischen Grundoperationen durch entsprechende Hardware direkt zugeordnet, wodurch sich eine einfache Programmierung ergibt.

Eine Vereinheitlichung des Befehlsumfanges von speicherprogrammierbaren Steuerungen ist zur Zeit nicht abzusehen. Daraus resultiert, daß die Programmiersprachen auf Betriebsebene entwickelt werden und nicht vereinheitlicht sind.

Die im folgenden aufgezeigten Befehlssätze und die daraus abgeleiteten Programmbeispiele können deshalb nur beispielhaft sein.

3.1. Programmerstellungen nach vorgegebenem Befehlssatz

Für eine einfache Steuerung ist folgender Befehlssatz denkbar:

Mnemonik		Operation
UND	X	Konjunktive Verknüpfung der Eingangsvariablen X mit dem Akkumulator Qu
UND	$\overline{X}$	Konjunktive Verknüpfung der negativen Eingangsvariablen X mit dem Akkumulator Qu
UND	Qo	Konjunktive Verknüpfung des Akkumulators Qo mit dem Akkumulator Qu
UND	$\overline{Qo}$	Konjunktive Verknüpfung des Akkumulators $\overline{Qo}$ mit dem Akkumulator Qu
ODER	X	Disjunktive Verknüpfung der Eingangsvariablen X mit dem Akkumulator Qo
ODER	$\overline{X}$	Disjunktive Verknüpfung der negierten Eingangsvariablen X mit dem Akkumulator Qo
ODER	Qu	Disjunktive Verknüpfung des Akkumulators Qu mit dem Akkumulator Qo
ODER	$\overline{Qu}$	Disjunktive Verknüpfung des Akkumulators $\overline{Qu}$ mit dem Akkumulator Qo
SP Qo Y		Speichern des Inhaltes von Qo in einen Ausgangsspeicher Y
SP $\overline{Qo}$ Y		Speichern des Inhaltes von Qo in einen Ausgangsspeicher Y
SP Qu Y		Speichern des Inhaltes von Qu in einen Ausgangsspeicher Y

SP $\overline{Qu}$ Y	Speichern des Inhalts von $\overline{Qu}$ in einen Ausgangsspeicher Y
JMP A	unbedingter Sprung zur Programmadresse A
JMP H A	bedingter Sprung bei high zur Programmadresse A
JMP L A	bedingter Sprung bei low zur Programmadresse A
LADE	Übernahme des Prozeßzustandes in den Eingangsspeicher

Bemerkung: Nach jeder Übernahme des Wertes aus den Akkumulatoren wird dieser in den Anfangszustand gesetzt (normiert), um die nachfolgende Verknüpfung ausführen zu können.

D. h. nach den Befehlen SP Qu; SP $\overline{Qu}$; ODER Qu und ODER $\overline{Qu}$ wird Qu = 1 gesetzt und nach den Befehlen SP Qo; SP $\overline{Qo}$; UND Qo und UND $\overline{Qo}$ wird Qo = 0 gesetzt.

Bei den Sprungbefehlen wird keine Normierung ausgeführt. Um die folgenden Programme übersichtlich zu halten, wird oft die Normierung durch Speicherbefehle nach Yo erreicht, wobei Yo keine Prozeßausgangsgröße ist. Ebenfalls ist der Eingang Xo stets mit low (0) belegt.

Mit diesem Befehlssatz kann die als Bool'sche Gleichung vorliegende Steuerungsaufgabe sehr günstig direkt in die Befehlsfolge umgesetzt werden.

Beispiel: $Y1 = X4\ \overline{X2}\ \overline{X6} \vee X4\ X2\ \overline{X1}$

$Y2 = X2$

$Y3 = X3 \vee \overline{X2}$

Die Zuordnung der Adressen der Variablen und Steuerwerte soll mit dem Index übereinstimmen, um übersichtliche Verhältnisse zu erhalten.

Das zugehörige Programm hat dann folgende Form:

Programmspeicheradresse	Mnemonik
00	LADE
01	UND X4
02	UND $\overline{X2}$
03	UND $\overline{X6}$
04	ODER Qu
05	UND X4
06	UND X2

07	UND $\overline{X1}$
08	ODER Qu
09	SP Qo Y1
10	UND X2
11	SP Qu Y2
12	ODER X3
13	ODER $\overline{X2}$
14	SP Qo Y3
15	JMP 00

Entsprechend den Vertauschungsregeln der Schaltalgebra kann sowohl innerhalb der konjunktiven Terme als auch die Reihenfolge der disjunktiv verknüpften Ausdrücke vertauscht werden. Ebenfalls ist für kombinatorische Gleichungssysteme die Reihenfolge beliebig.

Der Befehlssatz gestattet es, alle Gleichungen auch mit Sprungbefehlen zu programmieren. Das folgende Programm zeigt für Y1 die Vorgehensweise:

00	LADE
01	UND X4
02	JMP L 06
03	UND $\overline{X2}$
04	JMP L 06
05	UND $\overline{X6}$
06	ODER Qu
07	JMP H 14
08	UND X4
09	JMP L 13
10	UND X2
11	JMP L 13
12	UND $\overline{X1}$
13	ODER Qu
14	SP Qo Y1

Die Abarbeitungszeit wird sich gegenüber dem Programm mit logischen Befehlen verkürzen. Als Mittelwert kann gesagt werden, daß 50 % der Befehle ausgeführt werden. Der Speicherplatzbedarf erhöht sich allerdings in diesem Beispiel.

Eine Kombination beider Varianten wird sich also für die Praxis als sinnvoll erweisen. Wann Sprünge ausgeführt werden, ist von

der speziellen Aufgabe abhängig und muß jeweils abgeschätzt werden. Auch bei diesen bedingten Sprüngen kann die Reihenfolge der Testung der Variablen entsprechend den Regeln der Schaltalgebra vertauscht werden.

Berechnung von Zwischenfunktionen

Wiederkehrende Gleichungsteile in logischen Gleichungssystemen können einmal berechnet und in Zwischenspeichern abgelegt werden. Dadurch kann das mehrmalige Berechnen gleicher Teile entfallen. Grundsätzlich sollen die Zwischenfunktionen zuerst berechnet werden, um in einem Zyklus immer die gültigen Zwischenwerte weiterverrechnen zu können.

Der Ort, wo die Zwischenwerte abgespeichert werden, ist von der speziellen Struktur abhängig. Viele Steuerungen besitzen einen speziellen Zwischenspeicher, der durch seinen Adreßbereich von den Prozeßein- und -ausgängen abgrenzt und mit den Speicherbefehlen beschrieben werden kann. Eine weitere Verrechnung der Zwischenwerte kann mit den Befehlen, die Eingangsvariable verarbeiten, vorgenommen werden. Die Unterscheidung, ob Prozeß- oder Zwischenvariable, ist durch die Adresse gegeben. Der Ladebefehl wirkt nicht auf diese Speicher, so daß ein sofortiges Lesen nach der Berechnung möglich ist. In einfachsten Steuerungen ist kein Zwischenspeicher vorhanden. Hier kann durch die Verbindung von freien (für die Prozeßführung nicht benötigten) Ein- und Ausgängen eine Abfrage und damit eine weitere Verrechnung von Zwischenwerten erreicht werden. Dabei ist allerdings zu beachten, daß durch den Lade-Befehl die Reaktion erst im nächsten Zyklus erfolgen kann.

Beispiel: $Y1 = K1\,(X2 \vee \overline{X1})$ $\quad K1 = \overline{X4}\ X3 \vee X5$

$Y2 = K1\ \overline{X1X2}$

Für die Zwischenwerte ist ein Speicher vorhanden, der ab Adresse 100 genutzt werden kann.

Programmspeicheradresse	Mnemonik	
00	LADE	
01	UND $\overline{X4}$	
02	UND X3	
03	ODER Qu	
04	ODER X5	
05	SP Qo Y 100	
06	ODER X2	
07	ODER $\overline{X1}$	
08	UND Qo	
09	UND X100	
10	SP Qu Y1	
11	UND X1	
12	UND X2	
13	ODER $\overline{Qu}$)	Befehle zur
14	UND Qo)	Bildung der Termnegation $\overline{X1X2}$
15	UND X100	
16	SP Qu Y2	
17	JMP 00	

Würde die Berechnung von K1 am Programmende vorgenommen werden, könnte der Wert erst im nächsten Zyklus für Y1 und Y2 wirksam werden. Dabei können aber die Eingangsgrößen X1....X5 ebenfalls ihre Werte geändert haben, wodurch Fehler auftreten könnten. Bei der Berechnung von sequentiellen Gleichungen (Speichergleichungen) wird wie bei Zwischenfunktionen verfahren. Im folgenden Beispiel ist vorausgesetzt, daß die Steuerung keinen Zwischenspeicher besitzt.

Beispiel: $Y1 = X4X2\ (\overline{X3} \vee Y1)$

An der Steuerung sei Y1 mit X1 fest verdrahtet. Dadurch lautet die Gleichung jetzt

$$Y1 = X4X2\ (\overline{X3} \vee X1).$$

Die Programmierung bereitet keine Schwierigkeiten.

Man kann aber auch aus der Speichergleichung die Setz- Und Rück - setzfunktionen bilden und diese programmieren. Für das Beispiel gilt

$$SY1 = X4\ X2\ \overline{X3}$$

$$RY1 = \overline{X4\ X2} = \overline{X4} \vee \overline{X2}$$

Das entsprechende Programm kann dann wie folgt geschrieben werden:

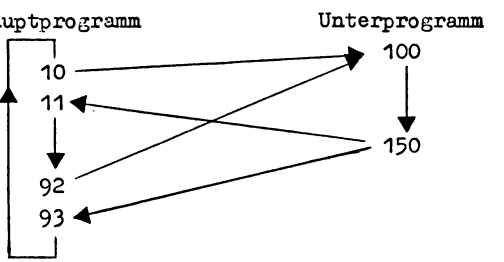
00	LADE	06	JMP 00
01	UND X4	07	ODER $\overline{X4}$
02	UND X2	08	ODER $\overline{X2}$
03	UND $\overline{X3}$	09	JMP L 00
04	JMP L 07	10	SP $\overline{Qo}$ Y1
05	SP Qu Y1	11	JMP 00

Dabei ist zu beachten, daß bei Nichterfüllung der S- bzw. R-Funktionen die Speicherbefehle zu überspringen sind, um den ursprünglichen Zustand nicht zu verändern.

Wenn die Befehlsliste bedingte und unbedingte Sprünge zuläßt, und die Möglichkeit der Zwischenspeicherung von Werten besteht, kann die Programmierung in einer Art Unterprogrammtechnik vorgenommen werden. Wiederkehrende Programmteile (Unterprogramme) können durch bedingte (prozeßzustandsabhängige) oder unbedingte Sprünge erreicht werden.

Problematisch wird der Rücksprung am Ende des Unterprogrammes, da kein Speicher vorhanden ist, der die Rücksprungadresse abspeichert. Dies ist lösbar, wenn jeder Absprungstelle ein 1-Bit-Zwischenspeicher zugeordnet wird, der vor der Absprungstelle gesetzt und nach Rückkehr rückgesetzt wird. Beim Rücksprung wird ein bedingter Sprung in Abhängigkeit dieses Zwischenspeichers ausgeführt. Das folgende Beispiel zeigt eine mögliche Verbindung von Haupt- und Unterprogramm.

Beispiel:



Das Unterprogramm sei auf den Programmspeicheradressen 100...150 untergebracht. Die Absprungstellen im Hauptprogramm sind auf den Adressen 10 und 92 willkürlich festgelegt. Es sei noch vereinbart, daß Xo mit low beschaltet ist. Das Programm kann folgende Form haben:

Hauptprogramm		Unterprogramm	
08	UND $\overline{Xo}$		
09	SP Qu Y14	100	⋮
10	JMP 100		⋮
11	UND Xo		⋮
12	SP Qu Y14	147	UND Y14
	⋮	148	JMP 1 11
90	UND $\overline{Xo}$	149	UND Y15
91	SP Qu Y15	150	JMP 1 93
92	JMP 100		
93	UND Xo		
94	SP Qu Y15		
	⋮		

Auf die gleiche Art kann mit einem Betriebsartenwahlschalter das Steuerregime für einen Prozeß vorgewählt werden. In Abhängigkeit von der Stellung des Wahlschalters werden Steuerprogramme entsprechend der Betriebsart abgearbeitet. Im "Hauptprogramm" wird dann nur noch die Wahlschalterstellung abgefragt.

3.2. Programmierung einer verbal vorgegebenen komplexen Aufgabe

Die zu lösende Aufgabe liegt dem Programmierer oft als verbale Beschreibung oder Ablaufschema vor. Seine Aufgabe liegt zunächst in der Formalisierung des Steuerungsproblems. Dies kann durch logische Gleichungen oder wie in der Datenverarbeitung mit Ablaufplänen geschehen. Bei der Verwendung logischer Befehle ist die Form der Bool'schen Gleichungen vorteilhaft. Wird mit Sprungbefehlen die logische Aufgabe gelöst, ist der Ablaufplan den Gleichungen vorzuziehen.

In beiden Fällen ist eine Zuordnung der Prozeßvariablen zu den Steuerungseingängen durch Festlegung der Adressen zu treffen. Gleiches gilt auch für die Steuerwerte und die Zwischenvariablen.

Ein etwas umfangreicheres Beispiel soll die Vorgehensweise darlegen:

Beispiel: Ein Flüssigkeitsbehälter kann durch zwei Pumpen gespeist werden. Die Entnahme der Flüssigkeit kann über vier Zapfstellen erfolgen, die durch Magnetventile geöffnet und geschlossen werden. Die Anforderungen für die Magnetventile werden über Taster eingegeben. Folgende Sensoren sind im Behälter untergebracht:

Füllstand Maximum
Füllstand Minimum
Temperatur Maximum (ϑmax)
Temperatur Minimum (ϑmin)

Steuerungsaufgabe:

Wird die Anforderungstaste für ein Magnetventil betätigt, so soll dieses geöffnet werden, solange nicht schon 3 Magnetventile geöffnet sind (das Öffnen von 4 Magnetventilen muß die Steuerung verhindern). Sind ein oder zwei Zapfstellen geöffnet, so soll Pumpe P1 Flüssigkeit in den Behälter nachfüllen. Bei drei geöffneten Zapfstellen wird noch zusätzlich Pumpe P2 eingeschaltet. Wird der obere Füllstand erreicht, so werden die Pumpen abgeschaltet. Ist der untere Füllstand erreicht, wird stets Pumpe P2 eingeschaltet. Bei Überschreitung des minimalen Füllstandes schaltet sich P2 wieder aus.

Zur Gewährleistung einer bestimmten Temperatur wird eine Heizung mit dem Signal ϑ min eingeschaltet und bei Erreichen des Wertes ϑ max ausgeschaltet.

Die Aufgabe ist mit einer speicherprogrammierbaren Steuerung zu lösen. Die Befehlsliste ist im Kapitel 3.1. gegeben.

1. Schritt Zuordnung der Variablen und Steuerwerte zu Adressen

Anforderung für MV 1 ≙ X1
" " MV 2 ≙ X2
" " MV 3 ≙ X3
" " MV 4 ≙ X4
Füllstand min ≙ X5
" max ≙ X6

ϑ min		≙ X7	Pumpe 1		≙ Y1
ϑ max		≙ X8	Pumpe 2		≙ Y2
Stellung	MV 1	≙ X9	Steuerwert	MV 1	≙ Y9
"	MV 2	≙ X10	"	MV 2	≙ Y10
"	MV 3	≙ X11	"	MV 3	≙ Y11
"	MV 4	≙ X12	"	MV 4	≙ Y12
"	Heizung	≙ X13	Heizung		≙ Y13

2. Schritt Formalisierung der Aufgabe und Vereinfachung

Das Zuschalten der Zapfstelle soll geschehen, wenn die Anforderung vorliegt und noch nicht drei andere Zapfstellen geöffnet sind. Die Gleichungen dafür lauten:

$$Y9 = X1\,\overline{X10X11X12} = X1\,(\overline{X10}\vee\overline{X11}\vee\overline{X12})$$
$$Y10 = X2\,\overline{X9X11X12} = X2\,(\overline{X9}\vee\overline{X11}\vee\overline{X12})$$
$$Y11 = X3\,\overline{X9X10X12} = X3\,(\overline{X9}\vee\overline{X10}\vee\overline{X12})$$
$$Y12 = X4\,\overline{X9X10X11} = X4\,(\overline{X9}\vee\overline{X10}\vee\overline{X11})$$

Die Pumpe P1 soll arbeiten, wenn ein oder zwei oder drei Zapfstellen geöffnet sind. Durch die obigen Bedingungen ist das Zuschalten von vier Zapfstellen verhindert. Für die Pumpengleichungen ergibt sich dadurch die Vereinfachung, daß die Kombination X9 X10 X11 X12 = 1 nie auftreten und mit $\emptyset$ belegt werden kann.

Daraus ergibt sich

$$Y1 = (X9\vee X10\vee X11\vee X12)\,\overline{X6}$$

Für Pumpe P2 gilt:

$$Y2 = (X9X10X11\vee X9X10X12\vee X9X11X12\vee X10X11X12)\,\overline{X6}\vee X5$$

Die Heizung kann mit einer sequentiellen Gleichung beschrieben werden.

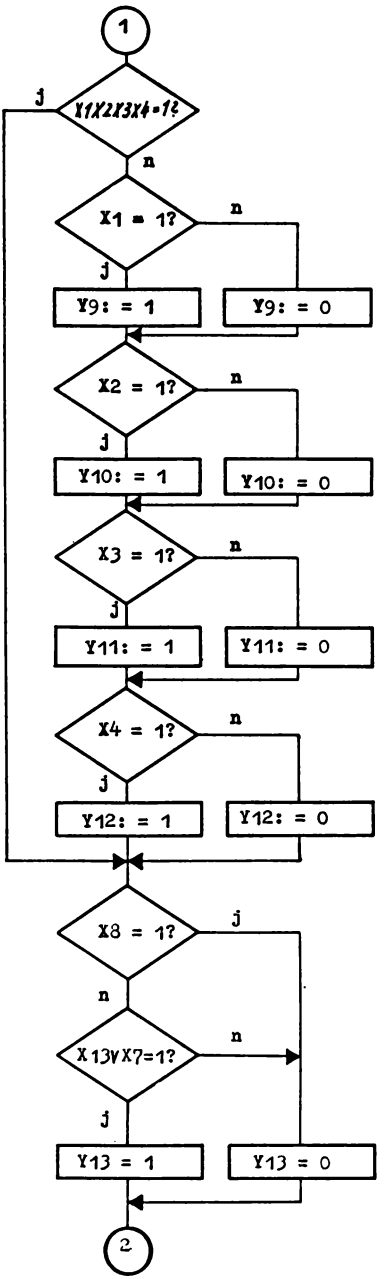
$$H = \vartheta\text{max}\,(H\vee\vartheta\text{min}) \text{ bzw.}$$
$$Y13 = \overline{X8}\,(X13\vee X7)$$

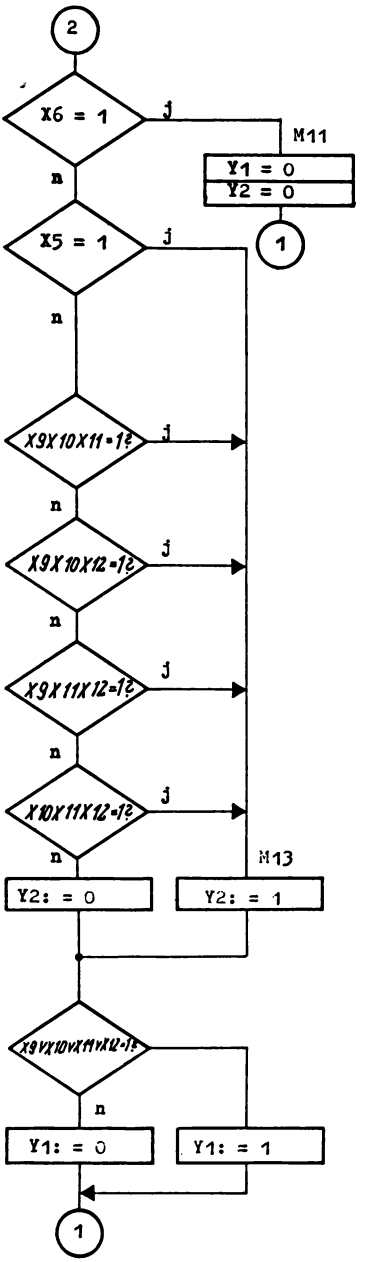
Die gefundenen Gleichungen werden programmiert.

00	LADE	29	UND Qo
01	ODER $\overline{X10}$	30	UND $\overline{X6}$
02	ODER $\overline{X11}$	31	SP Qu Y1
03	ODER $\overline{X12}$	32	UND X9
04	UND Qo	33	UND X10
05	UND X1	34	UND X11

```
06  SP Qu Y9          35  ODER Qu
07  ODER X̄9           36  UND X9
08  ODER X̄11          37  UND X10
09  ODER X̄12          38  UND X12
10  UND Qo            39  ODER Qu
11  UND X2            40  UND X9
12  SP Qu Y10         41  UND X11
13  ODER X̄9           42  UND X12
14  ODER X̄10          43  ODER Qu
15  ODER X̄12          44  UND X10
16  UND Qo            45  UND X11
17  UND X3            46  UND X12
18  SP Qu Y11         47  ODER Qu
19  ODER X̄9           48  UND Qo
20  ODER X̄10          49  UND X̄6
21  ODER X̄11          50  ODER Qu
22  UND Qo            51  ODER X5
23  UND X4            52  SP Qo Y2
24  SP Qo Y12         53  ODER X7
25  ODER X9           54  ODER X13
26  ODER X10          55  UND Qo
27  ODER X11          56  UND X̄8
28  ODER X12          57  SP Qu Y13
                      58  JMP 00
```

Der für die Programmierung mit Sprüngen günstigere Ablaufplan ist im folgenden für die gleiche Aufgabe dargestellt:


1
X1X2X3X4=1?
j
n
X1 = 1?
n
j
Y9: = 1
Y9: = 0
X2 = 1?
n
j
Y10: = 1
Y10: = 0
X3 = 1?
n
j
Y11: = 1
Y11: = 0
X4 = 1?
n
j
Y12: = 1
Y12: = 0
X8 = 1?
j
n
X13∨X7=1?
n
j
Y13 = 1
Y13 = 0
2


2
X6 = 1
j
M11
Y1 = 0
Y2 = 0
1
n
X5 = 1
j
n
X9X10X11=1?
j
n
X9X10X12=1?
j
n
X9X11X12=1?
j
n
X10X11X12=1?
j
n
M13
Y2: = 0
Y2: = 1
X9∨X10∨X11∨X12=1?
n
Y1: = 0
Y1: = 1
1

Das zum Programmablaufplan gehörende Programm hat die Form:
Xo = 0

```
00  LADE          39  JMP 1 61
01  UND X1        40  UND X9
02  UND X2        41  UND X10
03  UND X3        42  UND X11
04  UND X4        43  JMP 1 61
05  JMP 1 26      44  SP Qu Y0
06  UND X1        45  UND X9
07  JMP 0 10      46  UND X10
08  SP Qu Y09     47  UND X12
09  JMP 11        48  JMP 1 61
10  SP Qu Y09     49  SP Qu Y0
11  UND X2        50  UND X9
12  JMP 0 15      51  UND X11
13  SP Qu Y10     52  UND X12
14  JMP 16        53  JMP 1 61
15  SP Qu Y10     54  SP Qu Y0
16  UND X3        55  UND X10
17  JMP 0 20      56  UND X11
18  SP Qu Y11     57  UND X12
19  JMP 21        58  JMP 1 61
20  SP Qu Y11     59  SP Qu Y2
21  UND X4        60  JMP 62
22  JMP 0 25      61  SP Qu Y2
23  SP Qu Y12     62  SP Qo Y0
24  JMP 26        63  ODER X9
25  SP Qu Y12     64  ODER X10
26  ODER X8       65  ODER X11
27  JMP 1 31      66  ODER X12
28  ODER X7       67  JMP 1 70
29  ODER X13      68  SP Qo Y1
30  JMP 1 34      69  JMP 00
31  UND X0        70  UND X̄0
32  SP Qu Y13     71  SP Qu Y1
33  JMP 36        72  JMP 00
34  UND X0        73  UND X0
35  SP Q̄u Y13     74  SP Qu Y1
36  ODER X6       75  UND X0
37  JMP 1 73      76  SP Qu Y2
38  ODER X5       77  JMP 00
```

4. Die industrielle Steuerung PS 2000

Die programmierbare Steuerung PS 2000 wird serienmäßig von VEB Numerik "Karl-Marx" Karl-Marx-Stadt hergestellt. Sie ist einsetzbar in verschiedenen Industriezweigen (z.B. Maschinenbau, Polygraphie, Chemieindustrie, Textilindustrie ...).

Ihre wesentlichen Parameter sind:

Summe der E/A Signale	max.	512
Programmspeicherkapazität	max.	4095 Wörter
Anzahl der problemorientierten Befehle		20
Zykluszeit		≤ 18 ms
Anzahl der Peripheriebaugruppen	max.	32

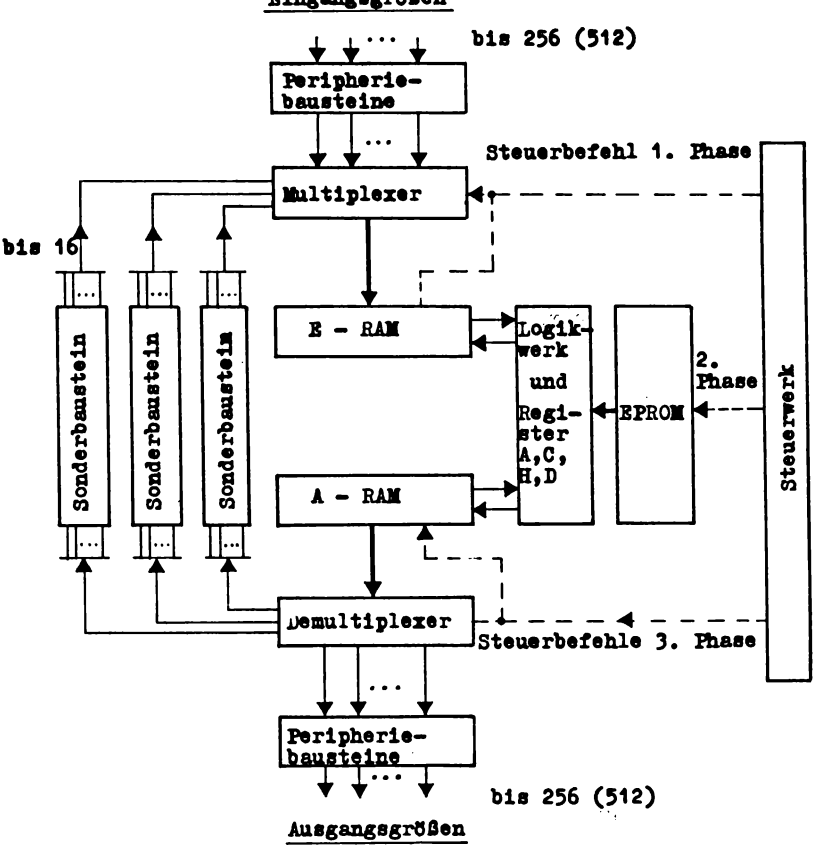


Abb. 30 Blockschaltbild der Steuerung PS 2000

Arbeitsweise der Steuerung:

Abb. 30 zeigt das Blockschaltbild der Steuerung. Das Steuerwerk organisiert die Programmverarbeitung in drei Phasen. In der ersten Phase werden die Prozeßeingänge in einen RAM-Speicher (E-RAM) übernommen.

Diese Phase entspricht dem Ladebefehl in der unter Punkt 3 beschriebenen Steuerung.

In der zweiten Phase wird das Anwenderprogramm abgearbeitet. Die berechneten Steuerwerte werden auf einen zweiten RAM-Speicher (A-RAM) abgespeichert.

In der dritten Phase wird der Inhalt des A-RAM s an die Ausgänge der Steuerung ausgegeben. Danach wird in die Phase 1 zurückgesprungen.

Die Programme für Phase 1 und Phase 3 sowie Programme zur Funktionsüberwachung werden vom Steuerungshersteller geliefert. Für die Programmierung der Steuerungsaufgabe stehen dem Nutzer folgende Befehle zur Verfügung:

LE LA	LEC LAC	UE UA	UEC UAC	E-RAM A-RAM
 TA	TEC	OE OA	OEC OAC	E-RAM A-RAM
positiv | negiert Transport		positiv | negiert Logik		

Der 1. Buchstabe gibt die Art der Operation an.

L = Laden des Akkumulators (AC)
T = Transport aus AC
U = UND-Verknüpfung mit AC
O = ODER-Verknüpfung mit AC

Der 2. Buchstabe gibt an, mit wem die Operation ausgeführt werden soll.

A = Ausgangsspeicher
E = Eingangsspeicher

Der 3. Buchstabe kennzeichnet die Abarbeitung des Complementes. Als Zusatzbefehle sind noch vorhanden:

END	bedeutet unbedingter Sprung zur Phase 3
NUL	keine Operation (Raum für Ergänzungen)
HS, HR	Hauptregister setzen bzw. rücksetzen
DS, DR	Direktregister setzen bzw. rücksetzen

Das Hauptregister kann zur Programmvereinfachung genutzt werden. Mit HS wird der Inhalt des AC in das Hauptregister geschrieben. Bei jedem folgenden Transportbefehl TA wird automatisch eine konjunktive Verknüpfung AC mit dem Hauptregisterinhalt zusätzlich ausgeführt.

Mit HR kann die Wirkung von HS aufgehoben werden.

Das Direktregister wird für schnelle Abarbeitungen genutzt, wenn der augenblickliche Zustand der Eingangsvariablen benötigt wird. Mit DS wird das Register gesetzt. Damit wird bewirkt, daß die durch entsprechende LE-Befehle aufgerufenen Variablen direkt vom Prozeß in den E-RAM gespeichert werden. Dadurch stehen die aktuellen Werte zur Abarbeitung bereit. Die TA-Befehle werden dahingehend erweitert, daß sofort ein Transport zu den Ausgangskarten angeschlossen wird.

Mit DR wird die Wirkung von DS aufgehoben.

Zwischen DS und DR dürfen die Befehle LEC, LA, LAC nicht geschrieben werden.

Der Befehlsvorrat gestattet eine ähnliche Programmierung wie in Kapitel 3 beschrieben.

Unterschiede sind:

- Die erste Variable einer Verknüpfung muß mit LADE eingeschrieben werden.
- Durch die fehlenden Sprungbefehle kann keine Verzweigung der Programme, keine Verkürzung der Zykluszeiten und keine Unterprogrammtechnik realisiert werden.

Die Steuerung PS 2000 besitzt Sonderbausteine zur Realisierung von Zeitverzögerungen, Zählaufgaben ....
Sie sind praktisch als Rückführungen zwischen Aus- und Eingänge geschaltet (siehe auch Abb. 30).

Die für die spezielle Aufgabe benötigten Ein- und Ausgänge kann der Anwender durch Einsatz von E- und A-Karten selbst wählen, wobei der Maximalwert 512 nicht überschritten werden darf.

Folgende E/A-Kartenbaugruppen stehen zur Verfügung:

Typ	Pegel	Bit-Baugr.	Belastung
Eingang	-12 V GS	16	15 mA
	-24 V GS		20 mA
	+24 V GS		
	- 48 V GS		
	+48 V GS		
	110 V WS		10 mA
Ausgang	potentialfrei		
	220 V (Relais)	16	0,4 A
	+24 V GS	6	3 A

Für die Programmeingabe und -testung stehen dem Nutzer ein E-PROM-Programmiergerät (Typ 1702 A) und ein Inbetriebnahmegerät (Typ IBG 2000) zur Verfügung. Der Herstellerbetrieb führt für Nutzer Anwenderschulungen durch, die Detailkenntnisse zum Einsatz, der Programmierung und Wartung vermitteln.

Literaturverzeichnis

/1/	Autorenkollektiv	Einführung in die Mikroelektronik für Erstanwender. Lehrbrief 1-3 KDT-Lehrmaterial, Berlin 1980
/2/	Schwarz, W./Meyer, G./ Eckhardt, D.	Mikrorechner Verlag Technik, Berlin 1980
/3/	Roth, M.	Mikroprozessoren Wissenschaftliche Zeitschrift TH Ilmenau 1980
/4/	Fischer, D./ Mauersberger, K.	Einführung in die digitale elektronische Steuerungstechnik. Lehrbrief 1-3 KDT-Lehrmaterial, Berlin 1980
/5/	Bechen, P.	Speicherprogrammierbare Steuerungen. Elektronik. 1980 H. 14, S. 41-53
/6/	Habiger, E./Hammer, V.	Mikroprogrammsteuerwerke - Eigenschaften und Anwendungsmöglichkeiten in der industriellen Steuerungstechnik msr 24 (1981) H 7, S. 370-375
/7/	Näser, P.	D 410- eine IS für industriellen Einsatz unter prozeßnahen Bedingungen rfe 1981 H. 10, S. 618-622
/8/	Weller, W.	Anwendung der Mikroelektronik in der Prozeßautomatisierung. Verlag Technik Berlin 1979, Reihe Automatisierungstechnik
/9/	Weller, W./Wilke, H.	Programmierbare Steuereinrichtungen Verlag Technik, Berlin 1981 Reihe Automatisierungstechnik